IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore® RELEASE 1.6

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review Quick Links » Sea

Welcome to IEEE Xplore®
- Home
- What Can I Access?
- Log-out

Tables of Contents
- Journals & Magazines
- Conference Proceedings
- Standards

Search
- By Author
- Basic
- Advanced

Member Services
- Join IEEE
- Establish IEEE Web Account
- Access the IEEE Member Digital Library

Your search matched **23** of **995179** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

(holistic or wholistic) <sentence> (handwriting or cha   Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **The role of holistic paradigms in handwritten word recognition**
Madhvanath, S.; Govindaraju, V.;
Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 23 , Issue: 2 , Feb. 2001
Pages:149 - 164

[Abstract] [PDF Full-Text (2408 KB)] **IEEE JNL**

---

2 **A survey of methods and strategies in character segmentation**
Casey, R.G.; Lecolinet, E.;
Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 18 , Issue: 7 , July 1996
Pages:690 - 706

[Abstract] [PDF Full-Text (1900 KB)] **IEEE JNL**

---

3 **Classical taxonomy: a holistic perspective of temperature measuring systems and instruments**
Henderson, I.A.; McGhee, J.;
Science, Measurement and Technology, IEE Proceedings A , Volume: 140 , Iss 4 , July 1993
Pages:263 - 268

[Abstract] [PDF Full-Text (360 KB)] **IEE JNL**

---

4 **Thai handwriting legal amounts recognition**
Chatwiriya, W.; Klinkhachorn, P.; Lass, N.;
System Theory, 2003. Proceedings of the 35th Southeastern Symposium on , 18 March 2003
Pages:381 - 385

[Abstract] [PDF Full-Text (407 KB)] **IEEE CNF**

---

5 **A shape based post processor for Gurmukhi OCR**
Lehal, G.S.; Singh, C.; Lehal, R.;
Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on , 10-13 Sept. 2001
Pages:1105 - 1109

[Abstract] [PDF Full-Text (424 KB)] **IEEE CNF**

---

6 **Evaluation of holistic features for word recognition**
de Oliveira, J.J.; de Carvalho, J.M.;
Computer Graphics and Image Processing, 2001 Proceedings of XIV Brazilian Symposium on , 15-18 Oct. 2001
Pages:394

[Abstract] [PDF Full-Text (60 KB)] **IEEE CNF**

---

7 **Holistic handwritten word recognition using discrete HMM and self-organizing feature map**
Dehghan, M.; Faez, K.; Ahmadi, M.; Shridhar, M.;
Systems, Man, and Cybernetics, 2000 IEEE International Conference on , Volu 4 , 8-11 Oct. 2000
Pages:2735 - 2739 vol.4

[Abstract] [PDF Full-Text (367 KB)] **IEEE CNF**

---

8 **Spectral features for Arabic word recognition**
Khorsheed, M.S.; Clocksin, W.F.;
Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 20 IEEE International Conference on , Volume: 6 , 5-9 June 2000
Pages:3574 - 3577 vol.6

[Abstract] [PDF Full-Text (276 KB)] **IEEE CNF**

---

9 **Whole word recognition in facsimile images**
Sherkat, N.; Allen, T.J.;
Document Analysis and Recognition, 1999. ICDAR '99. Proceedings of the Fift International Conference on , 20-22 Sept. 1999
Pages:547 - 550

[Abstract] [PDF Full-Text (36 KB)] **IEEE CNF**

---

10 **Empirical design of a holistic verifier for automatic sorting of handwritten Singapore postal addresses**
Chin Keong Lee; Leedham, G.;
Document Analysis and Recognition, 1999. ICDAR '99. Proceedings of the Fift International Conference on , 20-22 Sept. 1999
Pages:733 - 736

[Abstract] [PDF Full-Text (64 KB)] **IEEE CNF**

---

11 **Machine printed character segmentation method using side profiles**
Min-Chul Jung; Yong-Chul Shin; Srihari, S.N.;
Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. IEEE International Conference on , Volume: 6 , 12-15 Oct. 1999
Pages:863 - 867 vol.6

[Abstract] [PDF Full-Text (648 KB)] **IEEE CNF**

---

12 **Character recognition in practice today and tomorrow**
Miletzki, U.;
Document Analysis and Recognition, 1997., Proceedings of the Fourth Interna Conference on , Volume: 2 , 18-20 Aug. 1997
Pages:902 - 907 vol.2

[Abstract] [PDF Full-Text (652 KB)] **IEEE CNF**

---

13 **Recognising letters in on-line handwriting using hierarchical fuzzy inference**
Hennig, A.; Sherkat, N.; Whitrow, R.J.;
Document Analysis and Recognition, 1997., Proceedings of the Fourth Interna Conference on , Volume: 2 , 18-20 Aug. 1997
Pages:936 - 940 vol.2

[Abstract] [PDF Full-Text (448 KB)] **IEEE CNF**

---

14 **Pruning large lexicons using generalized word shape descriptors**
Madhvanath, S.; Krpasundar, V.;
Document Analysis and Recognition, 1997., Proceedings of the Fourth Interna Conference on , Volume: 2 , 18-20 Aug. 1997
Pages:552 - 555 vol.2

[Abstract] [PDF Full-Text (320 KB)] **IEEE CNF**

---

15 **Contour-based image preprocessing for holistic handwritten word recognition**
Madhvanath, S.; Govindaraju, V.;
Document Analysis and Recognition, 1997., Proceedings of the Fourth Interna Conference on , Volume: 2 , 18-20 Aug. 1997
Pages:536 - 539 vol.2

[Abstract] [PDF Full-Text (308 KB)] **IEEE CNF**

---

1 2 Next

Empirical design of a holistic verifier for automatic s of handwritten Singapore postal addresses

Chin Keong Lee Leedham, G.
Sch. of Appl. Sci., Nanyang Technol. Inst., Singapore;

This paper appears in: **Document Analysis and Recognition, 1999. ICDAR Proceedings of the Fifth International Conference on**

Meeting Date: 09/20/1999 - 09/22/1999
Publication Date: 20-22 Sept. 1999
Location: Bangalore India
On page(s): 733 - 736
Reference Cited: 5
Number of Pages: xxiv+821
Inspec Accession Number: 6352933

Abstract:
This paper describes an algorithmic architecture of an optical character recogn system integrated with a Singapore handwritten address interpretation (SHWA The proposed OCR-SHWAI system generates multiple hypotheses of postcode the hypotheses using a postal dictionary, and uses address features to choose hypotheses. The performance of this system on a set of 450 fictitious but real handwritten mail pieces shows an improvement in sorting performance from 8 correct, 6.0% reject, 12.4% error using OCR alone on the postcode to 70.9% 28.5% reject; 0.7% error representing a significant improvement in the error

Index Terms:
document image processing feature extraction handwritten character recognition opti recognition postal services OCR-SHWAI system Singapore handwritten address inte system address features algorithmic architecture automatic handwritten Singapore po sorting error rate fictitious handwritten mail pieces **holistic** verifier hypothesis verific multiple hypothesis generation optical **character recognition** system performance p dictionary realistic handwritten mail pieces

Documents that cite this document
There are no citing documents available in IEEE Xplore at this time.

Empirical Design of a Holistic Verifier for Automatic Sorting of Handwritten Singapore Postal Addresses

Chin Keong Lee and Graham Leedham
Nanyang Technological University
School of Applied Science, Intelligent System Laboratory
N4-#2A-36, Nanyang Avenue, Singapore 639798
Email: {pa2568530, asgleedham}@ntu.edu.sg

Abstract

This paper describes an algorithmic architecture of an optical character recognition (OCR) system integrated with a Singapore handwritten address interpretation (SHWAI) system. The proposed OCR-SHWAI system generates multiple hypotheses of postcodes; verifies the hypotheses using a postal dictionary; and the use of the address features to choose among the hypotheses. The performance of this system on a set of 450 fictitious but realistic hand-written mail pieces shows an improvement in sorting performance from 81.6% correct, 6.0% reject; 12.4% error using OCR alone on the postcode to 70.9% correct; 28.5% reject; 0.7% error representing a significant improvement in the error rate.

Keywords: dictionary look-up, handwritten address interpretation, features extraction, address verification.

1. Introduction

Automatic sorting of handwritten postal addresses in the absence of human intervention is a non-trivial task. Conventionally, the problem has been focused on locating and recognising the numeric fields such as postcode. Part of the address that contains redundant information such as the street names, building number, etc., have received less attention. Most of the recognition accuracy rates reported so far are over 95% [1] for individual numeric characters. However, improvement on the postcode recognition itself is insufficient especially when dealing with samples which are distorted or written in more 'personal' styles. The off-line handwritten word recognition (HWR) task is made difficult by the totally unconstrained, immense variation of handwriting styles and address formats, incomplete lexicons resulting from errors in postcode recognition and intrinsic deficiencies in the address database. Currently this problem is beyond the capabilities of handwriting recognition algorithms. Since sorting errors are costly and delay the service, maximum correct sorting and minimum error is the key requirement of an automatic sorting system.

In Singapore, a relatively simple postcode system exists. A fixed length of six digits code is used to address the country. The system allows Singapore Post Centre (SPC) to assign a unique postal code to every house and building in Singapore. The 6-digit postcode is made up of the sector code (first two digits) and four additional digits. The sector code defines a particular area in Singapore. The remaining four digits define a delivery point which can be a house or a building. The postal code or postcode is assigned based on two general categories, (a) Housing and Development Board (HDB) Residential Blocks and (b) Landed Property/ Condominiums/ Private Apartments/ Commercial Buildings/ HDB Industrial Blocks.

Even if the system for six digits postcode is equipped with a character recogniser with a performance of 98% recognition rate, which is comparable with the performance of human beings recognising characters without context, one out of six digits would be wrongly recognised which could lead to unacceptable mis-sorting. For example, for a 96.8% digit recogniser, the recognition rate for a six digits postcode is 0.98^6 = 88.6%, or an average error rate of 11.4%. In practice, a low error rate is essential, although a higher rejection rate can be tolerated because to sort rejected letters by hand is much cheaper than to later process wrongly destined letter mails.

A computer compatible database (Issue of January 1997) of postal addresses was obtained from SPC. It contains 115,494 unique postcodes and keys to identify street names and building names in corresponding files. Normally, the address consists of two or three lines depending on the type of address. The same address can also be written in more than three lines by different people but usually only the last three lines of an address is likely to contain information useful for sorting. Hence, the unpredictable structure of the hand-written address increased the difficulty of the problem.

2. Overview of the System Architecture

Figure 1 shows the overall system architecture of the proposed system:-

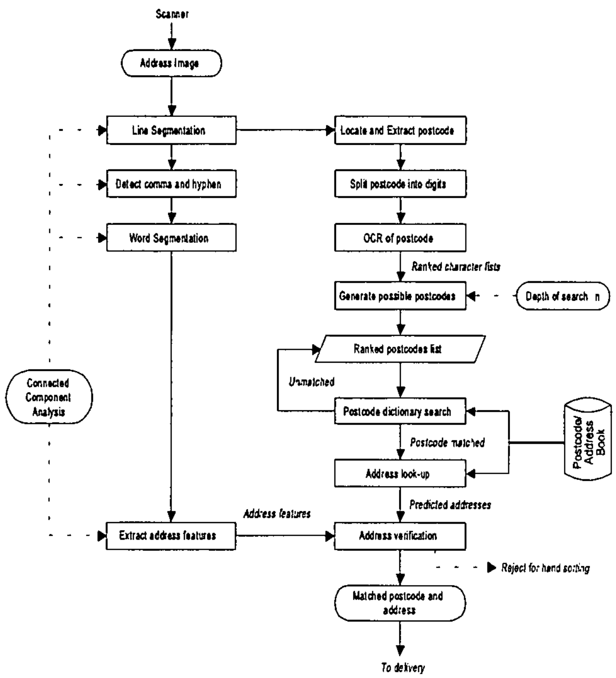


Figure 1. System architecture (OCR-SHWAI)

heuristic rules to the basic blocks and estimated centerlines such that every block was assigned a line number at the end of the process (Figure 2).

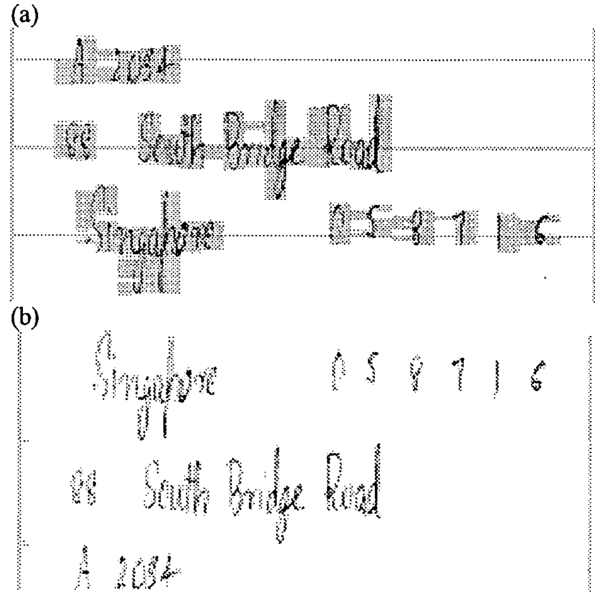


Figure 2. Segmenting address lines, (a) The effect of the shading technique. (b) The segmented lines.

2.1. Image Database Acquisition

Test binary images were obtained using a Hewlett Packard Scanjet flat bed scanner. Images were scanned at a resolution of 200 dpi in both axes for consistency with the resolution standards used on commercial postal sorting systems (e.g AEG systems). The address database was used to randomly generate a set of 900 fictitious but realistic postal addresses. These addresses had correct postcode, street name, building number and name (if any) but fictitious unit number for HDB flat.

The image database of handwritten postal addresses was obtained from a random population which comprised of University staff and student, outside-campus office workers, elderly and kids. The address was written on printed cards exhibiting four lightly coloured lines and six boxes into which the six-digit postcode were written. These lines and boxes dropped-out during the scanning process. The purpose of using constrained boxes was to isolate the character string segmentation which represents a sub-problem that can be resolved later.

2.2. Line Segmentation Process

The line segmentation process is trivial when no overlapping or touching lines exist. A direct method to separate this type of address quickly is by horizontal histogram. However, in the case of overlapping or touching lines, the line segmentation is accomplished by estimating the centerlines from a smoothed histogram and applying the shading technique [2] to build basic blocks. The overlapping lines were then segmented by applying heuristic rules to the basic blocks and estimated centerlines such that every block was assigned a line number at the end of the process (Figure 2).

2.3. Hypothesis Generation and Ranking Paradigm

A hypothesis generation and ranking (HGR) paradigm was used as the basis of a solution to the address verification strategy. The top five ranked confidence levels of character returned by the OCR algorithm were utilised to hypothesise and rank postcode verifications. The recognised postcode returned by the character recognition algorithm consists of 10 ranked confidence levels for each digit position. To determine how many choices to allow for each digit and thus how many possible postcodes, the cumulative probability of finding the correct digit within the ranked options was examined. From this, the first 5 (depth of search $n=5$) ranked choices at each digit position were considered, and thus a total of 15,625 possible postcodes was generated and ranked according to the confidence value which was determined by multiplying the confidence values of every digit.

2.4. Postcode Dictionary Look-up and Address Database Search

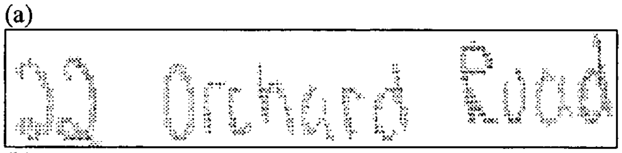
Each postcode from the ranked postcodes' list was retrieved sequentially and checked against a postcode dictionary for a valid entry. If a match exists, the predicted address corresponding to the valid hypothesis was retrieved from the address database by the address look-up module. The address features were extracted from the ASCII data and recorded in a linked list of address data structures. The searching continues with the next lower

ranked postcode in the list if the previous postcode was not found in the database. Together with the address image's features vector, this data structure was passed into the verification strategy module for postcode verification.

2.5. Word Segmentation Process

Word segmentation was performed for each text line of the address block using a horizontal smearing and special marks detection. Before horizontal smearing takes place, comma, hyphen and hash symbol detection were first applied to locate and remove the special marks and unit number if they exist. After smearing, each unique label of the components is then considered a word. The final stage is to split the over-smeared word if any, which has a length exceeding a threshold value (Figure 3).

(a)



(b)

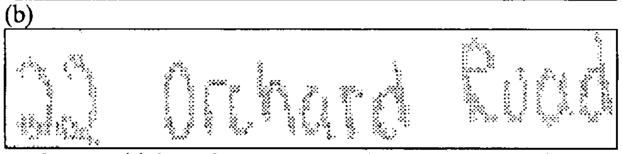


Figure 3. (a) Sorted components with a unique label for each component. (b) Words segmented after smearing

2.6. Address Features Extraction

The address features used for verification were: (i) global features - line and word positions and the number of words, (ii) local features - number of characters, loops, ascenders, descenders and ascenders/descenders sequence in each word were used. The line and word positions and number of words were extracted during the line and word segmentation processes.

In order to recognise the many variations of the same address, features that are invariant to certain transformations of the words need to be used. Invariants are features that have approximately the same values for samples of the same class that are, for example, addresses which are written in many different ways and with a large variation in human handwriting. However, not all variations among handwritten addresses from the same class (including noise or image degradation) can be modelled using invariants. If invariant features can not be found, an alternative is to pre-process the input image to improve the image quality, such as skew normalisation, baseline detection and correction, thining and so on. However, one should keep in mind that this introduces new discretization errors.

This set of features was used because they are invariant and most relevant for address similarity comparison purposes, in the sense of minimising the within-class pattern variability while enhancing the between-class pattern variability. However, features such as the initial character of the word, closed and open lobe location and stroke features may be considered in future. The phenomenon called the *curse of dimensionality* [3] cautions us that in the statistical classification approach, the number of features must be kept reasonably small if a limited training set is to be used.

2.7. Address Verification Strategy

The overall verification strategy is to extract global and local features from the words of the address image and to compare these against expected features of addresses corresponding to the postulated postcode(s). The presence or absence of expected features in the address as predicted from the postcode was used to improve or reduce respectively the confidence of the correctness of each hypothesised postcode.

To calculate the matching score between two addresses, all the degrees of similarity between words were recorded in the matching matrix. The rows of the matrix correspond with words from the reference address and its columns were associated with words from the address image. The problem now is to find the optimal combination of matches between the words from the address image and the reference address. The Mack's Bradford method [4] was chosen to solve this assignment problem. However, some modification must be made since the assignment problem deals with the problem of minimum criterion, and in this case we are dealing with a maximum criterion. Another constraint in the assignment problem was that it deals with a square matrix, but in our case the number of words in the address image and reference address might not be equal. The first problem can be overcome simply by multiplying all entries by -1 and proceeding to minimise. The second problem can be solved by assigning dummy columns or rows with zero entries, so that the matrix becomes square. The total matching score for the whole address image was then determined by applying aggregation operations on fuzzy sets, that is averaging operations, by which several fuzzy sets were combined to produce a single set $[0,1]^n \rightarrow [0,1]$ [5]. The total matching score represents how confident we are that this address is matched with the reference address. The nearer this value is to 1, the higher the confidence that the address is matched, and the nearer the value is to 0, the less confidence of the matched.

The Bradford method for the 2-dimensional assignment problem was proposed by Mack in 1969. The assignment problem is this: Given an $n \times n$ square array of numbers a_{ij}, define a 'possible' set as the set $a_{1,k_1}, a_{2,k_2}, \cdots, a_{n,k_n}$, where $k_1, k_2, \ldots k_n$ is a permutation of $1, 2, \ldots n$. Then find that possible set which has the minimum sum (or in some cases the maximum sum). Such sets are called 'minimal' or 'maximal'. Note that the set mentioned above consists of one element from each row but that no

two of them are in the same column (thus, since there are n of them they must be in separate columns). An example of a possible set is the set of diagonal elements i.e. $a_{11}, a_{22}, \cdots, a_{nn}$.

A variable N size buffer is used to store N number of the most possible hypotheses and the postcode which has the highest total matching score (TMS) in the address is selected. The decision is then made based on heuristic rules. If a sufficiently high match to expected features is found and ranked in first position the postcode is verified; else if the corresponding postcode was at a lower ranking but has an extremely high TMS value the postcode is recovered, otherwise the postcode is rejected.

3. Results and Performance Evaluation

The result of the performance evaluation of the (i) digit recogniser, (ii) raw OCR performance, (iii) OCR and dictionary look-up, and the (iv) cascaded OCR, dictionary look-up and verification strategy is tabulated in Table 1.

Table 1: Performance Evaluation Results

Rates (%)	Raw OCR performance		(iii) OCR and Dictionary Look-up			(iv) OCR, Dictionary Look-up & Verification (OCR-SHWAI)		
	(i) Digit	(ii) Post-code						
		total	TR	TE	total	TR	TE	total
Recog	96.8	81.6	89.1	84.9	87.2	71.5	70.9	71.4
Rej	0.6	6.0	2.7	7.2	4.7	28.1	28.5	28.0
Err	2.7	12.4	8.3	7.9	8.1	0.5	0.7	0.6

(TR = Training Set 450 images, TE = Testing Set 450 images)

There are 5,400 digits in 900 images (6 digit postcode) but due to some images which were non-OCR readable, only 5,232 digits were successfully fed into the OCR for recognition. The recognition rate of 96.8% was achieved by the neural net digit recogniser. The expected problem so-called "twins" error was observed in the "walk-up" type of address in which a unique postcode is assigned to every unit of houses that located along the same street. This "twins" phenomenon occurs when there are two or more postulated addresses that are identical in terms of their street names. The problem is that they each have a unique postcode. For example, when the "twins" addresses emerge as the best matched address and it is clear that only one of them has the correct postcode. The incorrect one is selected and will be mistaken as the correct postcode because it has the same address score as the real address and highest postcode confidence among themselves.

The loop feature has been used and partially overcame this problem by decreasing the total matching score of the fake addresses. However, the "twins" will remains unidentifiable when the block number of the addresses does not contain loops. A brute force way to overcome the "twins" error is by explicitly detect and recognise the building number but the disadvantage is that the alphanumeric characters may not be correctly recognised by the OCR algorithm. The "twins" phenomenon contributes 0.44% of errors and only a mere 0.11% error was due to intolerable features estimation errors. These give a total of 0.5% and 0.7% of errors in the TR and TE set respectively.

4. Conclusion

Automatic recognition and verification of handwritten addresses have been carried out in the United Kingdom, USA and Australia as well as in other countries but each country's postcode system has its own specific problems and solutions. This paper investigated the problems that hinder the automation of handwritten letter mail stream in Singapore addresses, and the techniques by which postcodes can be verified using the address knowledge. The main difference of the system from those used in other countries, such as British and United States, is its fixed length six digit numeric postal code system. There is only one central post office sorting centre, a smaller set of addresses, higher level of similarity in the structure of addresses and the entire postcode is verified against the address.

The raw postcode recognition produced high error rate of 12.4%. By adding a dictionary look-up, a relatively high postcode recognition rate (87.2%) was achieved but the error rate is still unacceptable (8.1%). The address verification strategy has achieved 0.6% error representing the major contribution of this research. Although, degradation was observed in the recognition rate, it is still very promising at 71.4%. The major problem of this postal code system is the high resolution of sorting, as a unique postcode can be assigned down to a building or house. This obstructs the feature-based verification strategy from differentiating the walk-up addresses. The rejection cases were mainly due to digit interference, incomplete or undetectable postcode, and postcode location error. Further work will focus on resolving the "twins" error found in the Singapore addresses to achieve higher reliability.

References

1. Suen, C. Y., Legault, R., Nadal, C., Cheriet, M., Lam, L., (1993), "Building a new generation of handwriting recognition systems," Pattern Recognition Letter, 14, 303-315.
2. Srihari, S. N., Cohen, E., Hull, J. J. and Kuan, L. (1989), "A system to locate and recognise zip codes in handwritten addresses," Int. Journal Of Research and Engineering-Postal Applications, 1, 37-45.
3. Jain, A. K. and Chandrasekaran, B. (1982), "Dimensionality and sample size considerations in pattern recognition practice," Handbook of Statistics, P. R. Krishnaiah and L. N. Kanal, eds, 2.
4. Mack, C. (1969), "The Bradford method for the assignment problem," The New J. of Stats. and Operational Research, 1, 17-29.
5. Klir, G. J. and Folger, T. A. (1988), "Fuzzy sets, Uncertainty and Information," Prentice-Hall Int., Inc., ISBN: 0133456382 (Int. ed.).

A survey of methods and strategies in character segmentation

Casey, R.G. Lecolinet, E.
IBM Almaden Res. Center, San Jose, CA, USA;

This paper appears in: **Pattern Analysis and Machine Intelligence, IEEE Transactions on**

Publication Date: July 1996
On page(s): 690 - 706
Volume: 18 , Issue: 7
ISSN: 0162-8828
Reference Cited: 89
CODEN: ITPIDJ
Inspec Accession Number: 5349782

Abstract:
Character segmentation has long been a critical area of the OCR process. The recognition rates for isolated characters vs. those obtained for words and conn character strings well illustrate this fact. A good part of recent progress in rea unconstrained printed and written text may be ascribed to more insightful han segmentation. This paper provides a review of these advances. The aim is to p appreciation for the range of techniques that have been developed, rather tha list sources. Segmentation methods are listed under four main headings. Wha termed the "classical" approach consists of methods that partition the input im subimages, which are then classified. The operation of attempting to decompo image into classifiable units is called "dissection." The second class of method dissection, and segments the image either explicitly, by classification of presp windows, or implicitly by classification of subsets of spatial features collected f image as a whole. The third strategy is a hybrid of the first two, employing dis together with recombination rules to define potential segments, but using clas select from the range of admissible segmentation possibilities offered by these subimages. Finally, **holistic** approaches that avoid segmentation by **recogniz character** strings as units are described

Index Terms:
hidden Markov models image segmentation optical **character recognition** OCR proc **character** segmentation connected **character** strings dissection **holistic** approaches

characters **recognition** rates unconstrained printed **words** written text

Documents that cite this document
Select link to view other documents in the database that cite this one.

Search Results [PDF FULL-TEXT 1900 KB] PREV NEXT DOWNLOAD CITATION

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

A Survey of Methods and Strategies in Character Segmentation

Richard G. Casey and Eric Lecolinet, *Member, IEEE*

Abstract—Character segmentation has long been a critical area of the OCR process. The higher recognition rates for isolated characters vs. those obtained for words and connected character strings well illustrate this fact. A good part of recent progress in reading unconstrained printed and written text may be ascribed to more insightful handling of segmentation.

This paper provides a review of these advances. The aim is to provide an appreciation for the range of techniques that have been developed, rather than to simply list sources. Segmentation methods are listed under four main headings. What may be termed the "classical" approach consists of methods that partition the input image into subimages, which are then classified. The operation of attempting to decompose the image into classifiable units is called "dissection." The second class of methods avoids dissection, and segments the image either explicitly, by classification of prespecified windows, or implicitly by classification of subsets of spatial features collected from the image as a whole. The third strategy is a hybrid of the first two, employing dissection together with recombination rules to define potential segments, but using classification to select from the range of admissible segmentation possibilities offered by these subimages. Finally, holistic approaches that avoid segmentation by recognizing entire character strings as units are described.

Index Terms—Optical character recognition, character segmentation, survey, holistic recognition, Hidden Markov Models, graphemes, contextual methods, recognition-based segmentation.

1 INTRODUCTION

1.1 The Role of Segmentation in Recognition Processing

Character segmentation is an operation that seeks to decompose an image of a sequence of characters into subimages of individual symbols. It is one of the decision processes in a system for optical character recognition (OCR). Its decision, that a pattern isolated from the image is that of a character (or some other identifiable unit), can be right or wrong. It is wrong sufficiently often to make a major contribution to the error rate of the system.

In what may be called the "classical" approach to OCR, Fig. 1, segmentation is the initial step in a three-step procedure:

Given a starting point in a document image:

1) Find the next character image.
2) Extract distinguishing attributes of the character image.
3) Find the member of a given symbol set whose attributes best match those of the input, and output its identity.

This sequence is repeated until no additional character images are found.

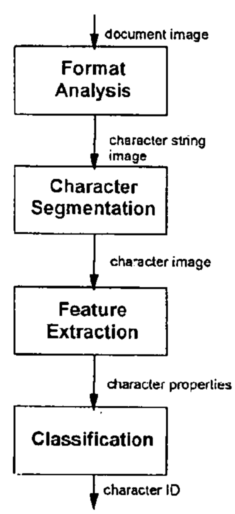


Fig. 1. Classical optical character recognition (OCR). The process of character recognition consists of a series of stages, with each stage passing its results on to the next in pipeline fashion. There is no feedback loop that would permit an earlier stage to make use of knowledge gained at a later point in the process.

An implementation of step 1, the segmentation step, requires answering a simply posed question: "What constitutes a character?" The many researchers and developers who have tried to provide an algorithmic answer to this question find themselves in a Catch-22 situation. A character is a pattern that resembles one of the symbols the system is designed to recognize. But to determine such a resem-

• *R.G. Casey is with the IBM Almaden Research Center, San Jose, CA 95120. E-mail: casey@ix.netcom.com.*
• *E. Lecolinet is with École Nationale Supérieure des Télécommunications (ENST), Paris, France.*

Manuscript received Mar. 18, 1995; revised Mar. 22, 1996. Recommended for acceptance by R. Kasturi.
For information on obtaining reprints of this article, please send e-mail to: transpami@computer.org, and reference IEEECS Log Number P96058.

0162-8828/96$05.00 ©1996 IEEE

blance the pattern must be segmented from the document image. Each stage depends on the other, and in complex cases it is paradoxical to seek a pattern that will match a member of the system's recognition alphabet of symbols without incorporating detailed knowledge of the structure of those symbols into the process.

Furthermore, the segmentation decision is not a local decision, independent of previous and subsequent decisions. Producing a good match to a library symbol is necessary, but not sufficient, for reliable recognition. That is, a poor match on a later pattern can cast doubt on the correctness of the current segmentation/recognition result. Even a series of satisfactory pattern matches can be judged incorrect if contextual requirements on the system output are not satisfied. For example, the letter sequence "cl" can often closely resemble a "d," but usually such a choice will not constitute a contextually valid result.

Thus, it is seen that the segmentation decision is interdependent with local decisions regarding shape similarity, and with global decisions regarding contextual acceptability. This sentence summarizes the refinement of character segmentation processes in the past 40 years or so. Initially, designers sought to perform segmentation as per the "classical" sequence listed above. As faster, more powerful electronic circuitry has encouraged the application of OCR to more complex documents, designers have realized that step 1 can not be divorced from the other facets of the recognition process.

In fact, researchers have been aware of the limitations of the classical approach for many years. Researchers in the 1960s and 1970s observed that segmentation caused more errors than shape distortions in reading unconstrained characters, whether hand- or machine-printed. The problem was often masked in experimental work by the use of databases of well-segmented patterns, or by scanning character strings printed with extra spacing. In commercial applications, stringent requirements for document preparation were imposed. By the beginning of the 1980s, workers were beginning to encourage renewed research interest [73] to permit extension of OCR to less constrained documents.

The problems of segmentation persist today. The well-known tests of commercial printed text OCR systems by the University of Nevada, Las Vegas [64], [65], consistently ascribe a high proportion of errors to segmentation. Even when perfect patterns, the bitmapped characters that are input to digital printers, were recognized, commercial systems averaged 0.5% spacing errors. This is essentially a segmentation error by a process that attempts to isolate a word subimage. The article [6] emphatically illustrates the woes of current machine-print recognition systems as segmentation difficulties increase (see Fig. 2). The degradation in performance of NIST tests of handwriting recognition on segmented [86] and unsegmented [88] images underscore the continuing need for refinement and fresh approaches in this area. On the positive side of the ledger, the study [29] shows the dramatic improvements that can be obtained when a thoroughgoing segmentation scheme replaces one of prosaic design.

Some authors previously have surveyed segmentation, often as part of a more comprehensive work, e.g., cursive recognition [36], [19], [20], [55], [58], [81], or document analysis [23], [29]. In the present paper, we present a survey whose focus is character segmentation, and which attempts to provide broad coverage of the topic.

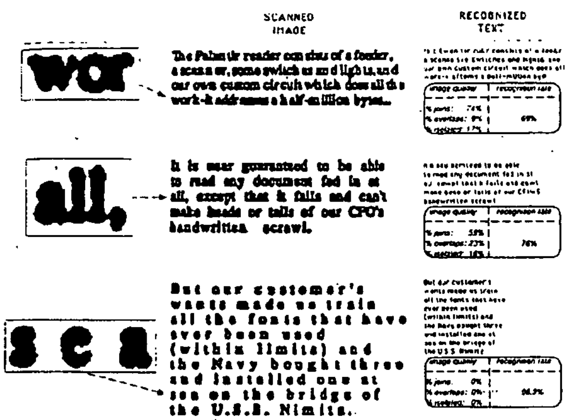


Fig. 2. Segmentation problems in machine print OCR. The figure illustrates performance of a 1991 prototype of a commercial reader as character spacing is varied. Proceeding from top to bottom, the images (at left) correspond to condensed, normal, and expanded spacing, respectively. Note that recognition output (at right) for the well-spaced images is quite accurate, but degrades badly when characters begin to merge. This system has continued to evolve and the illustration is not representative of performance of later releases. (Adapted from [6].)

1.2 Organization of Methods

A major problem in discussing segmentation is how to classify methods. Tappert et al. [81], for example, speaks of "external" vs. "internal" segmentation, depending on whether recognition is required in the process. Dunn and Wang [20] use "straight segmentation" and "segmentation-recognition" for a similar dichotomization.

A somewhat different point of view is proposed in this paper. The division according to use or nonuse of recognition in the process fails to make clear the fundamental distinctions among present-day approaches. For example, it is not uncommon in text recognition to use a spelling corrector as a post-processor. This stage may propose the substitution of two letters for a single letter output by the classifier. This is in effect a use of recognition to resegment the subimage involved. However, the process represents only a trivial advance on traditional methods that segment independent of recognition.

In this paper, the distinction between methods is based on how segmentation and classification interact in the overall process. In the example just cited, segmentation is done in two stages, one before and one after image classification. Basically an unacceptable recognition result is re-examined and modified by a (implied) resegmentation. This is a rather "loose" coupling of segmentation and classification.

A more profound interaction between the two aspects of recognition occurs when a classifier is invoked to select the segments from a set of possibilities. In this family of approaches, segmentation and classification are integrated. To

some observers it even appears that the classifier performs segmentation since, conceptually at least, it could select the desired segments by exhaustive evaluation of all possible sets of subimages of the input image.

After reviewing available literature, we have concluded that there are three "pure" strategies for segmentation, plus numerous hybrid approaches that are weighted combinations of these three. The elemental strategies are:

1) The classical approach, in which segments are identified based on "character-like" properties. This process of cutting up the image into meaningful components is given a special name, "dissection," in discussions below.
2) Recognition-based segmentation, in which the system searches the image for components that match classes in its alphabet.
3) Holistic methods, in which the system seeks to recognize words as a whole, thus, avoiding the need to segment into characters.

In strategy 1, the criterion for good segmentation is the agreement of general properties of the segments obtained with those expected for valid characters. Examples of such properties are height, width, separation from neighboring components, disposition along a baseline, etc. In method 2, the criterion is recognition confidence, perhaps including syntactic or semantic correctness of the overall result. Holistic methods (strategy 3) in essence revert to the classical approach with words as the alphabet to be read. The reader interested to obtain an early illustration of these basic techniques may glance ahead to Fig. 6 for examples of dissection processes, Fig. 13 for a recognition-based strategy, and Fig. 16 for a holistic approach.

Although examples of these basic strategies are offered below, much of the literature reviewed for this survey reports a blend of methods, using combinations of dissection, recognition searching, and word characteristics. Thus, although the paper necessarily has a discrete organization, the situation is perhaps better conceived as in Fig. 3. Here, the three fundamental strategies occupy orthogonal axes: Hybrid methods can be represented as weighted combinations of these lying at points in the intervening space. There is a continuous space of segmentation strategies rather than a discrete set of classes with well-defined boundaries. Of course, such a space exists only conceptually; it is not meaningful to assign precise weights to the elements of a particular combination.

Taking the fundamental strategies as a base, this paper is organized as indicated in Fig. 4. In Section 2, we discuss methods that are highly weighted towards strategy 1. These approaches perform segmentation based on general image features and then classify the resulting segments. Interaction with classification is restricted to reprocessing of ambiguous recognition results.

In Section 3, we present methods illustrative of recognition-based segmentation, strategy 2. These algorithms avoid early imposition of segmentation boundaries. As Fig. 4 shows, such methods fall into two subcategories. Windowing methods segment the image blindly at many boundaries chosen without regard to image features, and then try to choose an optimal segmentation by evaluating the classification of the subimages generated. Feature-based methods detect the physical location of image features, and seek to segment this representation into well-classified subsets. Thus, the former employs recognition to search for "hard" segmentation boundaries, the latter for "soft" (i.e., implicitly defined) boundaries.

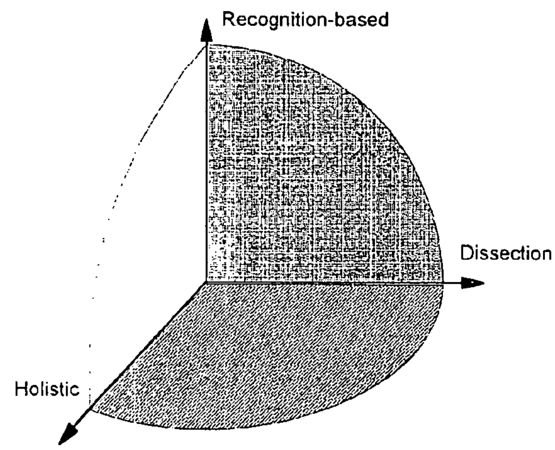


Fig. 3. The space of segmentation strategies. The fundamental segmentation strategies identified in the text are shown as occupying orthogonal axes. Many methods adopted in practice employ elements of more than one basic strategy. Although it is impossible to assign precise coordinates in this space, in concept such methods lie in the interior of the space bounded by the three planes shown.

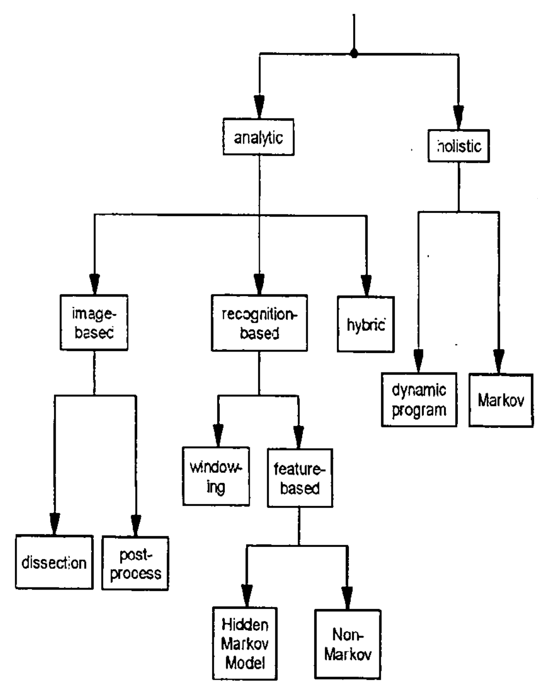


Fig. 4. Hierarchy of segmentation methods. The text follows the classification scheme shown.

Sections 2 and 3 describe contrasting strategies: one in which segmentation is based on image features, and a second in which classification is used to select from segmentation candidates generated without regard to image content. In Section 4, we discuss a hybrid strategy in which a preliminary segmentation is implemented, based on image features, but a later combination of the initial segments is performed and evaluated by classification in order to choose the best segmentation from the various hypotheses offered.

The techniques in Sections 2 to 4 are letter-based and, thus, are not limited to a specific lexicon. They can be applied to the recognition of any vocabulary. In Section 5, are presented holistic methods, which attempt to recognize a word as a whole. While avoiding the character segmentation problem, they are limited in application to a predefined lexicon. Markov models appear frequently in the literature, justifying further subclassification of holistic and recognition-based strategies, as indicated in Fig. 4. Section 6 offers some remarks and conclusions of the state of research in the segmentation area. Except when approaches of a sufficiently general nature are discussed, the discussion is limited to Western character sets as well as to "off-line" character recognition, that is to segmentation of character data obtained by optical scanning rather than at inception ("on-line") using tablets, light pens, etc. Nevertheless, it is important to realize that much of the thinking behind advanced work is influenced by related efforts in speech and on-line recognition, and in the study of human reading processes.

2 Dissection Techniques for Segmentation

Methods discussed in this section are based on what will be termed *"dissection"* of an image. By *dissection* is meant the decomposition of the image into a sequence of subimages using general features (as, for example, in Fig. 5). This is opposed to later methods that divide the image into subimages independent of content. Dissection is an intelligent process in that an analysis of the image is carried out; however, classification into symbols is not involved at this point.

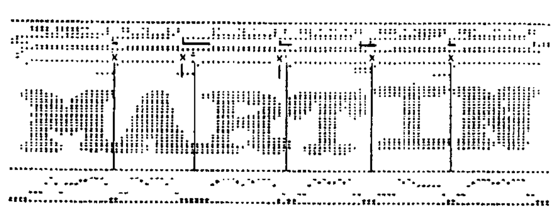


Fig. 5. The dissection method of Hoffman and McCullough. An evaluation function based on a running count of horizontal black-white and white-black transitions is plotted below the image. The horizontal black bars above the image indicate the activation region for the function. Vertical lines indicate human estimates of optimal segmentation (from [43]) columns, while Xs indicate columns chosen by the algorithm.

In literature describing systems where segmentation and classification do not interact, dissection is the entire segmentation process: The two terms are equivalent. However, in many current studies, as we shall see, segmentation is a complex process, and there is a need for a term such as "dissection" to distinguish the image-cutting subprocess from the overall segmentation, which may use contextual knowledge and/or character shape description.

2.1 Dissection Directly into Characters

In the late 1950s and early 1960s, during the earliest attempts to automate character recognition, research was focused on the identification of isolated images. Workers mainly sought methods to characterize and classify character shapes. In some cases, individual characters were mapped onto grids and pixel coordinates entered on punched cards [40], [49] in order to conduct controlled development and testing. As CRT scanners and magnetic storage became available, well-separated characters were scanned, segmented by dissection based on whitespace measurement, and stored on tape. When experimental devices were built to read actual documents they dealt with constrained printing or writing that facilitated segmentation.

For example, bank check fonts were designed with strong leading edge features that indicated when a character was properly registered in the rectangular array from which it was recognized [24]. Handprinted characters were printed in boxes that were invisible to the scanner, or else the writer was constrained in ways that aided both segmentation and recognition. A very thorough survey of status in 1961 [79] gives only implicit acknowledgment of the existence of the segmentation problem. Segmentation is not shown at all in the master diagram constructed to accompany discussion of recognition stages. In the several pages devoted to preprocessing (mainly thresholding), the function is indicated only peripherally as part of the operation of registering a character image.

The twin facts that early OCR development dealt with constrained inputs, while research was mainly concerned with representation and classification of individual symbols, explains why segmentation is so rarely mentioned in pre-70s literature. It was considered a secondary issue.

2.1.1 White Space and Pitch

In machine printing, vertical whitespace often serves to separate successive characters. This property can be extended to handprint by providing separated boxes in which to print individual symbols. In applications such as billing, where document layout is specifically designed for OCR, additional spacing is built into the fonts used. The notion of detecting the vertical white space between successive characters has naturally been an important concept in dissecting images of machine print or handprint.

In many machine print applications involving limited font sets, each character occupies a block of fixed width. The pitch, or number of characters per unit of horizontal distance, provides a basis for estimating segmentation points. The sequence of segmentation points obtained for a given line of print should be approximately equally spaced at the distance corresponding to the pitch. This provides a global basis for segmentation, since separation points are not independent.

Applying this rule permits correct segmentation in cases where several characters along the line are merged or broken. If most segmentations can be obtained by finding columns of white, the regular grid of intercharacter boundaries can be estimated. Segmentation points not lying near these boundaries can be rejected as probably due to broken characters. Segmentation points missed due to merged characters can be estimated as well, and a local analysis conducted in order to decide where best to split the composite pattern.

One well-documented early commercial machine that dealt with a relatively unconstrained environment was the reader IBM installed at the U.S. Social Security Administration in 1965 [38]. This device read alphanumeric data typed by employers on forms submitted quarterly to the SSA. There was no way for SSA to impose constraints on the printing process. Typewriters might be of any age or condition, ribbons in any state of wear, and the font style might be one or more of approximately 200.

In the SSA, reader segmentation was accomplished in two scans of a print line by a flying-spot scanner. On the initial scan, from left to right, the character pitch distance D was estimated by analog circuitry. On the return scan, right to left, the actual segmentation decisions were made using parameter D. The principal rule applied was that a double white column triggered a segmentation boundary. If none was found within distance D, then segmentation was forced.

Hoffman and McCullough [43] generalized this process and gave it a more formal framework (see Fig. 5). In their formulation, the segmentation stage consisted of three steps:

1) Detection of the start of a character.
2) A decision to begin testing for the end of a character (called sectioning).
3) Detection of end-of-character.

Sectioning, step 2, was the critical step. It was based on a weighted analysis of horizontal black runs completed, versus runs still incomplete as the print line was traversed column-by-column. An estimate of character pitch was a parameter of the process, although in experiments it was specified for 12-character per inch typewriting. Once the sectioning algorithm indicated a region of permissible segmentation, rules were invoked to segment based on either an increase in bit density (start of a new character) or else on special features designed to detect end-of-character. The authors experimented with 80,000 characters in 10- and 12-pitch serif fonts containing 22% touching characters. Segmentation was correct to within one pixel about 97% of the time. The authors noted that the technique was heavily dependent on the quality of the input images, and tended to fail on both very heavy or very light printing.

2.1.2 Projection Analysis

The vertical projection (also called the "vertical histogram") of a print line, Fig. 6a, consists of a simple running count of the black pixels in each column. It can serve for detection of white space between successive letters. Moreover, it can indicate locations of vertical strokes in machine print, or regions of multiple lines in handprint. Thus, analysis of the projection of a line of print has been used as a basis for segmentation of noncursive writing. For example, in [66], in segmenting Kanji handprinted addresses, columns where the projection fell below a predefined threshold were candidates for splitting the image.

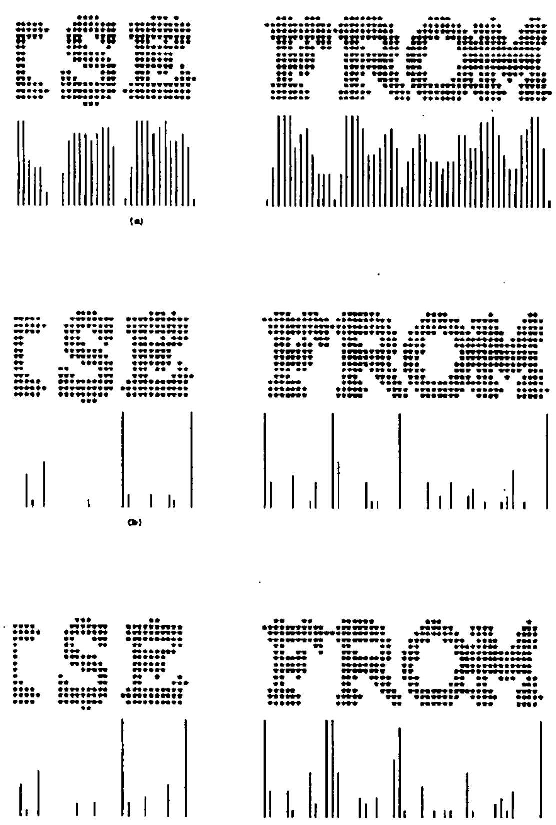


Fig. 6. Dissection based on projection: (a) Vertical projection of an image. It is an easy matter to detect white columns between characters, and regions of light contact. The function fails, however, to make clear the O-M separation. (b) Differencing measure for column splitting. The function from [1] is based on a second difference of the projection. This gives a clear peak at most separation columns, but may still fail for the O-M case. (c) Differencing measure after column ANDing. The image transformed by an AND of adjacent columns, with the difference function of (b) applied to the transformed image. The AND operation tends to increase separation, leading to a better defined peak at the O-M boundary.

When printed characters touch, or overlap horizontally, the projection often contains a minimum at the proper segmentation column. In [1], the projection was first obtained, then the ratio of second derivative of this curve to its height was used as a criterion for choosing separating columns (see Fig. 6b). This ratio tends to peak at minima of the projection, and avoids the problem of splitting at points along thin horizontal lines.

A peak-to-valley function was designed to improve on this method in [59]. A minimum of the projection is located and the projection value noted. The sum of the differences

between this minimum value and the peaks on each side is calculated. The ratio of the sum to the minimum value itself (plus 1, presumably to avoid division by zero) is the discriminator used to select segmentation boundaries. This ratio exhibits a preference for low valley with high peaks on both sides.

A prefiltering was implemented in [83] in order to intensify the projection function. The filter ANDed adjacent columns prior to projection as in Fig. 6c. This has the effect of producing a deeper valley at columns where only portions of the vertical edges of two adjacent characters are merged.

A different kind of prefiltering was used in [57] to sharpen discrimination in the vicinity of holes and cavities of a composite pattern. In addition to the projection itself, the difference between upper and lower profiles of the pattern was used in a formula analogous to that of [1]. Here, the "upper profile" is a function giving the maximum y-value of the black pixels for each column in the pattern array. The lower profile is defined similarly on the minimum y-value in each column.

A vertical projection is less satisfactory for the slanted characters commonly occurring in handprint. In one study [28], projections were performed at two-degree increments between –16 and +16 degrees from the vertical. Vertical strokes and steeply angled strokes such as occur in a letter A were detected as large values of the derivative of a projection. Cuts were implemented along the projection angle. Rules were implemented to screen out cuts that traversed multiple lines, and also to rejoin small floating regions, such as the left portion of a T crossbar, that might be created by the cutting algorithm. A similar technique is employed in [89].

2.1.3 Connected Component Processing

Projection methods are primarily useful for good quality machine printing, where adjacent characters can ordinarily be separated at columns. A one-dimensional analysis is feasible in such a case.

However, the methods described above are not generally adequate for segmentation of proportional fonts or handprinted characters. Thus, pitch-based methods can not be applied when the width of the characters is variable. Likewise, projection analysis has limited success when characters are slanted, or when inter-character connections and character strokes have similar thickness.

Segmentation of handprint or kerned machine printing calls for a two-dimensional analysis, for even nontouching characters may not be separable along a single straight line. A common approach (see Fig. 7) is based on determining connected black regions ("connected components," or "blobs"). Further processing may then be necessary to combine or split these components into character images.

There are two types of followup processing. One is based on the "bounding box," i.e., the location and dimensions of each connected component. The other is based on detailed analysis of the images of the connected components.

2.1.3.1 Bounding Box Analysis

The distribution of bounding boxes tells a great deal about the proper segmentation of an image consisting of noncursive characters. By testing their adjacency relationships to perform merging, or their size and aspect ratios to trigger splitting mechanisms, much of the segmentation task can be accurately performed at a low cost in computation.

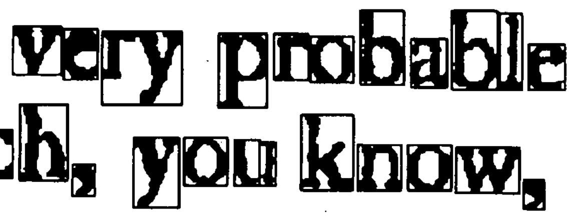


Fig. 7. Connected components. This example illustrates characters that consist of two components (e.g., the "u" in "you"), as well as components consisting of more than one character (e.g., the "ry" in "very"). The bounding box of each component is also shown. The latter is often used as a basis for dissection methods, as discussed in the text.

This approach has been applied, for example, in segmenting handwritten postcodes [14] using knowledge of the number of symbols in the code: six for the Canadian codes used in experiments. Connected components were joined or split according to rules based on height and width of their bounding boxes. The rather simple approach correctly classified 93% of 300 test codes, with only 2.7% incorrect segmentation and 4.3% rejection.

Connected components have also served to provide a basis for the segmentation of scanned handwriting into words [74]. Here, it is assumed that words do not touch, but may be fragmented. Thus, the problem is to group fragments (connected components) into word images. Eight different distance measures between components were investigated. The best methods were based on the size of the white run lengths between successive components, with a reduction factor applied to estimated distance if the components had significant horizontal overlap. 90% correct word segmentation was achieved on 1,453 postal address images.

An experimental comparison of character segmentation by projection analysis vs. segmentation by connected components is reported in [87]. Both segmenters were tested on a large data base (272,870 handprinted digits) using the same follow-on classifier. Characters separated by connected component segmentation resulted in 97.5% recognition accuracy, while projection analysis (along a line of variable slope) yielded almost twice the errors at 95.3% accuracy. Connected component processing was also carried out four time faster than projection analysis, a somewhat surprising result.

2.1.3.2 Splitting of Connected Components

Analysis of projections or bounding boxes offers an efficient way to segment nontouching characters in hand- or machine-printed data. However, more detailed processing is necessary in order to separate joined characters reliably. The intersection of two characters can give rise to special

image features. Consequently dissection methods have been developed to detect these features and to use them in splitting a character string image into subimages. Such methods often work as a follow-on to bounding box analysis. Only image components failing certain dimensional tests are subjected to detailed examination.

Another concern is that separation along a straight-line path can be inaccurate when the writing is slanted, or when characters are overlapping. Accurate segmentation calls for an analysis of the shape of the pattern to be split, together with the determination of an appropriate segmentation path (see Fig. 8).

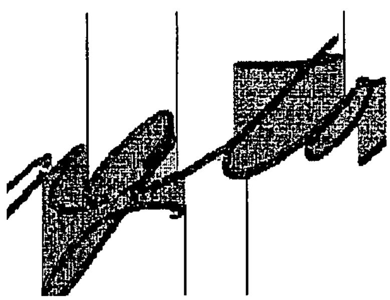


Fig. 8. Dissection based on search and deflect. The initial path of the cut trajectory is along a column corresponding to a minimum in the upper profile of the image. When black is encountered the path is modified recursively, seeking better positions at which to enforce a cut. In this way, multiple cuts can be made at positions which are in the shadow region of the image.

Depending upon the application, certain a priori knowledge may be used in the splitting process. For example, an algorithm may assume that some but not all input characters can be connected. In an application such as form data, the characters may be known to be digits or capital letters, placing a constraint on dimensional variations. Another important case commercially is handwritten zip codes, where connected strings of more than two or three characters are rarely encountered, and the total number of symbols is known.

These different points have been addressed by several authors. One of the earliest studies to use contour analysis for the detection of likely segmentation points was reported in [69]. The algorithm, designed to segment digits, uses local vertical minima encountered in following the bottom contour as "landmark points." Successive minima detected in the same connected component are assumed to belong to different characters, which are to be separated. Consequently, contour following is performed from the leftmost minimum counter-clockwise until a turning point is found. This point is presumed to lie in the region of intersection of the two characters and a cut is performed vertically. An algorithm that not only detects likely segmentation points, but also computes an appropriate segmentation path, as in Fig. 8, was proposed in [76] for segmenting digit strings. The operation comprises two steps. First, a vertical scan is performed in the middle of an image assumed to contain two possibly connected characters. If the number of black to white transitions is 0 or 2, then the digits are either not connected or else simply connected, respectively, and can therefore be separated easily by means of a vertical cut. If the number of transitions found during this scan exceeds 2, the writing is probably slanted and a special algorithm using a "Hit and Deflect Strategy" is called. This algorithm is able to compute a curved segmentation path by iteratively moving a scanning point. This scanning point starts from the maximum peak in the bottom profile of the lower half of the image. It then moves upwards by means of simple rules which seek to avoid cutting the characters until further movement is impossible. In most cases, only one cut is necessary to separate slanted characters that are simply connected.

This scheme was refined in a later technique [51], [52], which determines not only "how" to segment characters but also "when" to segment them. Detecting which characters have to be segmented is a difficult task that has not always been addressed. One approach consists in using recognition as a validation of the segmentation phase and resegmenting in case of failure. Such a strategy will be addressed in Section 3. A different approach, based on the concept of prerecognition, is proposed in [51].

The basic idea of the technique is to follow connected component analysis with a simple recognition logic whose role is not to label characters but rather to detect which components are likely to be single, connected or broken characters. Splitting of an image classified as connected is then accomplished by finding characteristic landmarks of the image that are likely to be segmentation points, rejecting those that appear to be situated within a character, and implementing a suitable cutting path.

The method employs an extension of the Hit and Deflect scheme proposed in [76]. First, the valleys of the upper and lower profiles of the component are detected. Then, several possible segmentation paths are generated. Each path must start from an upper or lower valley. Several heuristic criteria are considered for choosing the "best path" (the path must be "central" enough, paths linking an upper and a lower valley are preferred, etc.).

The complete algorithm works as a closed loop system, all segmentation being proposed and then confirmed or discarded by the prerecognition module: Segmentation can only take place on components identified as connected characters by prerecognition, and segmentations producing broken characters are discarded. The system is able to segment n-tuplets of connected characters which can be multiply connected or even merged. It was first applied on zip code segmentation for the French Postal Service.

In [85], an algorithm was constructed based on a categorization of the vertexes of stroke elements at contact points of touching numerals. Segmentation consists in detecting the most likely contact point among the various vertexes proposed by analysis of the image, and performing a cut similar in concept to that illustrated in Fig. 8.

Methods for defining splitting paths have been examined in a number of other studies as well. The algorithm of [17] performs background analysis to extract the face-up and face-down valleys, strokes and loop regions of component images. A "marriage score matrix" is then used to decide which pair of valleys is the most appropriate. The

separating path is deduced by combining three lines respectively segmenting the upper valley, the stroke and the lower valley.

A distance transform is applied to the input image in [31] in order to compute the splitting path. The objective is to find a path that stays as far from character strokes as possible without excessive curvature. This is achieved by employing the distance transform as a cost function, and using complementary heuristics to seek a minimal-cost path.

A shortest-path method investigated in [84] produces an "optimum" segmentation path using dynamic programming. The path is computed iteratively by considering successive rows in the image. A one-dimensional cost array contains the accumulated cost of a path emanating from a predetermined starting point at the top of the image to each column of the current row. The costs to reach the following row are then calculated by considering all vertical and diagonal moves that can be performed from one point of the current row to a point of the following row (a specific cost being associated to each type of move). Several tries can be made from different starting points. The selection of the best solution is based on classification confidence (which is obtained using a neural network). Redundant shortest-path calculations are avoided in order to improve segmentation speed.

2.1.3.3 Landmarks

In recognition of cursive writing, it is common to analyze the image of a character string in order to define lower, middle and upper zones. This permits the ready detection of ascenders and descenders, features that can serve as "landmarks" for segmentation of the image. This technique was applied to on-line recognition in pioneering work by Frischkopf and Harmon [36]. Using an estimate of character width, they dissected the image into patterns centered about the landmarks, and divided remaining image components on width alone. This scheme does not succeed with letters such as "u," "n," "m," which do not contain landmarks. However, the basic method for detecting ascenders and descenders has been adopted by many other researchers in later years.

2.2 Dissection with Contextual Postprocessing: Graphemes

The segmentation obtained by dissection can later be subjected to evaluation based on linguistic context, as shown in [7]. Here, a Markov model is postulated to represent splitting and merging as well as misclassification in a recognition process. The system seeks to correct such errors by minimizing an edit distance between recognition output and words in a given lexicon. Thus, it does not directly evaluate alternative segmentation hypotheses, it merely tries to correct poorly made ones. The approach is influenced by earlier developments in speech recognition. A non-Markovian system reported in [12] uses a spell-checker to correct repeatedly made merge and split errors in a complete text, rather than in single words as above.

An alternative approach still based on dissection is to divide the input image into subimages that are not necessarily individual characters. The dissection is performed at stable image features that may occur within or between characters, as for example, a sharp downward indentation can occur in the center of an "M" or at the connection of two touching characters. The preliminary shapes, called "graphemes" or "pseudo-characters" (see Fig. 9), are intended to fall into readily identifiable classes. A contextual mapping function from grapheme classes to symbols can then complete the recognition process. In doing so, the mapping function may combine or split grapheme classes, i.e., implement a many-to-one or one-to-many mapping. This amounts to an (implicit) resegmentation of the input. The dissection step of this process is sometimes called "presegmentation" or, when the intent is to leave no composite characters, "over-segmentation."

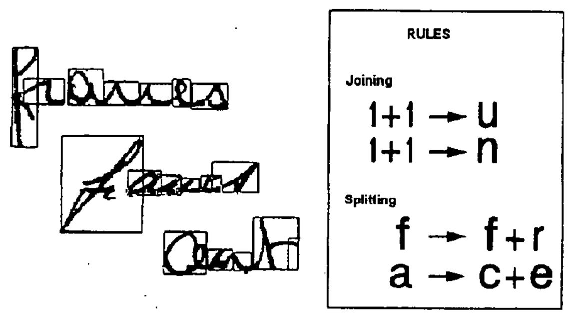


Fig. 9. Graphemes. The bounding boxes indicate subimages isolated by a cutting algorithm. The algorithm typically cuts "u" or "n" into two pieces as shown, but this is anticipated by the contextual recombination rules at right. Other rules are shown for cases where the cutting algorithm failed to split two characters. These rules are applied in many combinations seeking to transform the grapheme classes obtained during recognition into a valid character string.

The first reported use of this concept was probably in [72], a report on a system for off-line cursive script recognition. In this work, a dissection into graphemes was first performed based on the detection of characteristic areas of the image. The classes recognized by the classifier did not correspond to letters, but to specific shapes that could be reliably segmented (typically combinations of letters, but also portions of letters). Consequently, only 17 nonexclusive classes were considered.

As in [72], the grapheme concept has been applied mainly to cursive script by later researchers. Techniques for dissecting cursive script are based on heuristic rules derived from visual observation. There is no "magic" rule and it is not feasible to segment all handwritten words into perfectly separated characters in the absence of recognition. Thus, word units resulting from segmentation are not only expected to be entire characters, but also parts or combinations of characters (the graphemes). The relationship between characters and graphemes must remain simple enough to allow definition of an efficient post-processing stage. In practice, this means that a single character decomposes into at most two graphemes, and conversely, a single grapheme represents at most a two- or three-character sequence.

The line segments that form connections between characters in cursive script are known as "ligatures." Thus,

some dissection techniques for script seek "lower ligatures," connections near the baseline that link most lowercase characters. A simple way to locate ligatures is to detect the minima of the upper outline of words. Unfortunately, this method leaves several problems unresolved:

- Letters "o," "b," "v," and "w" are usually followed by "upper" ligatures.
- Letters "u" and "w" contain "intraletter ligatures," i.e., a subpart of these letters cannot be differentiated from a ligature in the absence of context.
- Artifacts sometimes cause erroneous segmentation.

In typical systems, these problems are treated at a later contextual stage that jointly treats both segmentation and recognition errors. Such processing is included in the system since cursive writing is often ambiguous without the help of lexical context. However, the quality of segmentation still remains very much dependent on the effectiveness of the dissection scheme that produces the graphemes.

Dissection techniques based on the principle of detecting ligatures were developed in [22], [61], and [53]. The last study was based on a dual approach:

- The detection of possible presegmentation zones.
- The use of a "prerecognition" algorithm, whose aim was not to recognize characters, but to evaluate whether a subimage defined by the presegmenter was likely to constitute a valid character.

Presegmentation zones were detected by analyzing the upper and lower profiles and open concavities of the words. Tentative segmentation paths were defined in order to separate words into isolated graphemes. These paths were chosen to respect several heuristic rules expressing continuity and connectivity constraints. However, these presegmentations were only validated if they were consistent with the decisions of the prerecognition algorithm. An important property of this method was independence from character slant, so that no special preprocessing was required.

A similar presegmenter was presented in [42]. In this case, analysis of the upper contour, and a set of rules based on contour direction, closure detection, and zone location were used. Upper contour analysis was also used in [47] for a presegmentation algorithm that served as part of the second stage of a hybrid recognition system. The first stage of this system also implemented a form of the hit and deflect strategy previously mentioned.

A technique for segmenting handwritten strings of variable length, was described in [27]. It employs upper and lower contour analysis and a splitting technique based on the hit and deflect strategy.

Segmentation can also be based on the detection of minima of the lower contour as in [8]. In this study, presegmentation points were chosen in the neighborhood of these minima and emergency segmentation performed between points that were highly separated. The method requires handwriting to be previously deslanted in order to ensure proper separation.

A recent study which aims to locate "key letters" in cursive words employs background analysis to perform letter segmentation [18]. In this method, segmentation is based on the detection and analysis of faceup and facedown valleys and open loop regions of the word image.

3 Recognition-Based Segmentation

Methods considered here also segment words into individual units (which are usually letters). However, the principle of operation is quite different. In principle, no feature-based dissection algorithm is employed. Rather, the image is divided systematically into many overlapping pieces without regard to content. These are classified as part of an attempt to find a coherent segmentation/recognition result. Systems using such a principle perform "recognition-based" segmentation: Letter segmentation is a by-product of letter recognition, which may itself be driven by contextual analysis. The main interest of this category of methods is that they bypass the segmentation problem: No complex "dissection" algorithm has to be built and recognition errors are basically due to failures in classification. The approach has also been called "segmentation-free" recognition. The point of view of this paper is that recognition necessarily involves segmentation, explicit or implicit though it be. Thus, the possibly misleading connotations of "segmentation-free" will be avoided in our own terminology.

Conceptually, these methods are derived from a scheme in [48] and [11] for the recognition of machine-printed words. The basic principle is to use a mobile window of variable width to provide sequences of tentative segmentations which are confirmed (or not) by character recognition. Multiple sequences are obtained from the input image by varying the window placement and size. Each sequence is assessed as a whole based on recognition results.

In recognition-based techniques, recognition can be performed by following either a serial or a parallel optimization scheme. In the first case, e.g., [11], recognition is done iteratively in a left-to-right scan of words, searching for a "satisfactory" recognition result. The parallel method [48] proceeds in a more global way. It generates a lattice of all (or many) possible feature-to-letter combinations. The final decision is found by choosing an optimal path through the lattice.

The windowing process can operate directly on the image pixels, or it can be applied in the form of weightings or groupings of positional feature measurements made on the images. Methods employing the former approach are presented in Section 3.1, while the latter class of methods is explored in Section 3.2.

Word level knowledge can be introduced during the recognition process in the form of statistics, or as a lexicon of possible words, or by a combination of these tools. Statistical representation, which has become popular with the use of Hidden Markov Models (HMMs), is discussed in Section 3.2.1.

3.1 Methods that Search the Image

Recognition-based segmentation consists of the following two steps:

1) Generation of segmentation hypotheses (windowing step).
2) Choice of the best hypothesis (verification step).

How these two operations are carried out distinguishes the different systems.

As easy to state as these principles are, they were a long time in developing. Probably the earliest theoretical and experimental application of the concept is reported by Kovalevsky [48]. The task was recognition of typewritten Cyrillic characters of poor quality. Although character spacing was fixed, Kovalevski's model assumed that the exact value of pitch and the location of the origin for the print line were known only approximately. He developed a solution under the assumption that segmentation occurred along columns. Correlation with prototype character images was used as a method of classification.

Kovalevsky's model (Fig. 10) assumes that the probability of observing a given version of a prototype character is a spherically symmetric function of the difference between the two images. Then the optimal objective function for segmentation is the sum of the squared distances between segmented images and matching prototypes. The set of segmented images that minimizes this sum is the optimal segmentation. He showed that the problem of finding this solution can be formulated as one of determining the path of maximum length in a graph, and that this path can be found by dynamic programming. This process was implemented in hardware to produce a working OCR system.

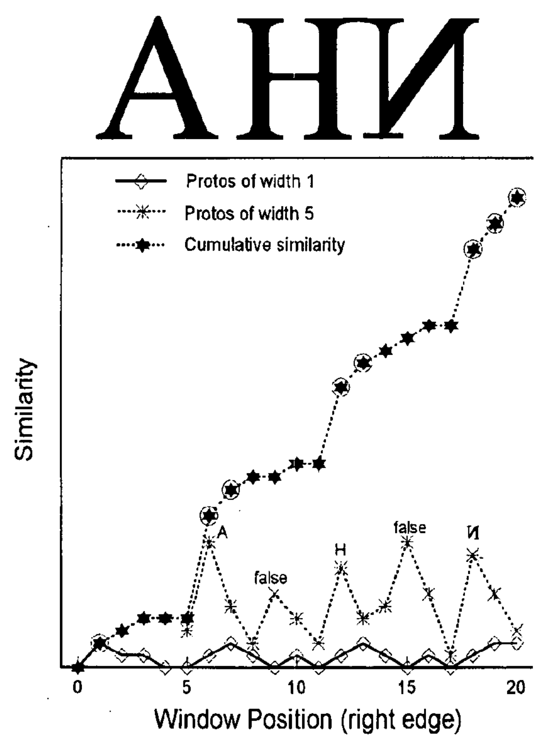


Fig. 10. Kovalevsky's method. The lower two curves give the similarities obtained by comparing stored prototypes against a window whose right edge is placed on the input image (shown at top) at the positions indicated by the abcissa. For window size 1 the only prototype is a blank column. For windows of size 5 there are multiple character prototypes, and the highest similarity among these is plotted. The cumulative plot gives the maximum total similarity obtainable from a sequence of windows up to the plotted point. Several good matches are obtainable by windowing parts of two characters. However, such "false" matches are not part of the optimal sequence of windows, indicated by the circled points of the upper graph.

Kovalevsky's work appears to have been neglected for some time. A number of years later, [11] reported a recursive splitting algorithm for machine-printed characters. This algorithm, also based on prototype matching, systematically tests all combinations of admissible separation boundaries until it either exhausts the set of cutpoints, or else finds an acceptable segmentation (see Fig. 11). An acceptable segmentation is one in which every segmented pattern matches a library prototype within a prespecified distance tolerance.

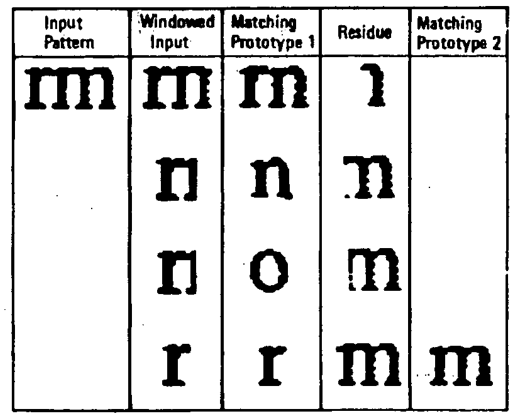


Fig. 11. Recursive segmentation. The example shows the results of applying windows of decreasing width to the left side of an input image. When the subimage in the window is recognized (in this case by matching a prototype character stored in the system's memory), then the procedure is recursively applied to the residue image. Recognition (and segmentation) is accomplished if a complete series of matching windows is found. In the top three rows, no match is obtained for the residue image, but successful segmentation is finally obtained as shown at the bottom.

A technique combining dynamic programming and neural net recognition was proposed in [10]. This technique, called "Shortest Path Segmentation," selects the optimal consistent combination of cuts from a predefined set of windows. Given this set of candidate cuts, all possible "legal" segments are constructed by combination. A graph whose nodes represent acceptable segments is then created and these nodes are connected when they correspond to compatible neighbors. The paths of this graph represent all the legal segmentations of the word. Each node of the graph is then assigned a "distance" obtained by the neural net recognizer. The shortest path through the graph thus corresponds to the best recognition and segmentation of the word.

The method of "selective attention" [30] takes neural networks even further in the handling of segmentation problems. In this approach, Fig. 12, a neural net seeks recognizable patterns in an image input, but is inhibited automatically after recognition in order to ignore the region of the recognized character and search for new character images in neighboring regions.

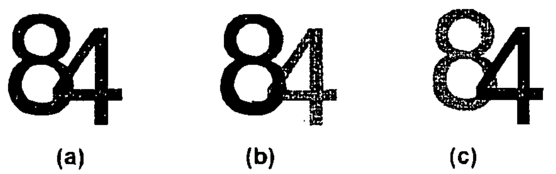


Fig. 12. Selective attention. (a) An input pattern. (b) The recognizer gradually reinforces pixels that corresponds to objects in its template library, and inhibits those that do not, yielding a partial recognition. (c) After a delay, attention is switched to unmatched regions, and another match to the library is found (after Fukishima).

3.2 Methods that Segment a Feature Representation of the Image

3.2.1 Hidden Markov Models

A Hidden Markov Model (often abbreviated HMM) models variations in printing or cursive writing as an underlying probabilistic structure which is not directly observable. This structure consists of a set of states plus transition probabilities between states. In addition, the observations that the system makes on an image are represented as random variables whose distribution depends on the state. These observations constitute a sequential feature representation of the input image. The survey [34] provides an introduction to its use in recognition applications.

For the purpose of this survey, three levels of underlying Markov model are distinguished, each calling for a different type of feature representation:

1) The Markov model represents letter-to-letter variations of the language. Typically such a model is based on bigram frequencies (first order model) or trigram frequencies (second order model). The features are gathered on individual characters or graphemes, and segmentation must be done in advance by dissection. Such systems are included in Section 2 above.
2) The Markov model represents state-to-state transitions within a character. These transitions provide a sequence of observations on the character. Features are typically measured in the left-to-right direction. This facilitates the representation of a word as a concatenation of character models. In such a system, segmentation is (implicitly) done in the course of matching the model against a given sequence of feature values gathered from a word image. That is, it decides where one character model leaves off and the next one begins, in the series of features analyzed. Examples of this approach are given in this section.
3) The Markov model represents the state-to-state variations within a specific word belonging to a lexicon of admissible word candidates. This is a holistic model as described in Section 5, and entails neither explicit or implicit segmentation into characters.

In this section, we are concerned with HMMs of type 2, which model sequences of feature values obtained from individual letters. For example, Fig. 13 shows a sample feature vector produced from the word "cat." This sequence can be segmented into three letters in many different ways, of which two are shown. The probability that a particular segmentation resulted from the word "cat" is the product of the probabilities of segment 1 resulting from "c," segment 2 from "a," etc. The probability of a different lexicon word can likewise be calculated. To choose the most likely word from a set of alternatives the designer of the system may select either the composite model that gives the segmentation having greatest probability, or else that model which maximizes the a posteriori probability of the observations, i.e., the sum over all segmentations. In either case, the optimization algorithm is organized to avoid redundant calculations, in the former case, by using the well-known Viterbi algorithm.

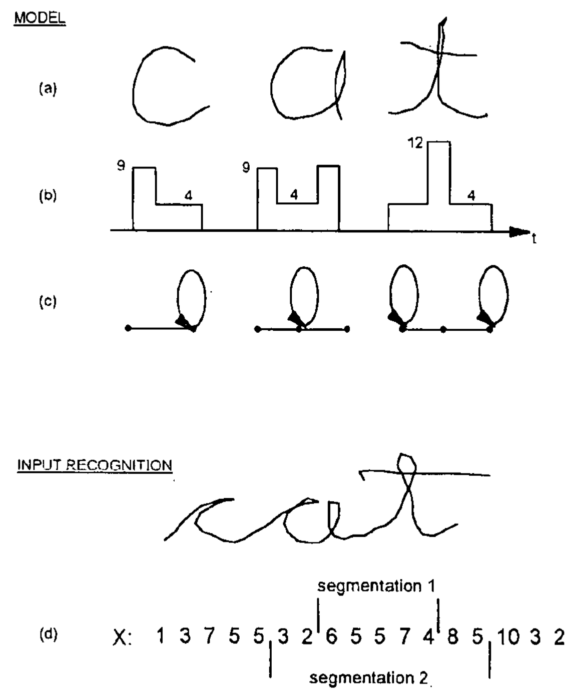


Fig. 13. Hidden Markov Models. This vastly oversimplified example is intended only to illustrate the principal concepts: (a) Training letters and (b) typical sequences of feature values obtained from (a). (c) The Markov models underlying (b), indicating the state sequence, and showing that certain states may be re-entered. Each state outputs a value from a feature distribution whose mean is indicated in the diagram above. The model for a word is the concatenation of such letter models. (d) A sequence of feature values obtained from a word, indicating several different segmentations. The HMM solution is found by evaluating many possible segmentation, of which two are shown.

Such HMMs are a powerful tool to model the fact that letters do not always have distinct segmentation boundaries. It is clear that in the general case perfect letter dissection can not be achieved. This problem can be compensated by the HMMs, as they are able to learn by observing letter segmentation behavior on a training set. Context (word and letter frequencies, syntactic rules) can be also be included, by defining transition probabilities between letter states.

Elementary HMMs describing letters can be combined to form either several model-discriminant word HMMs or else a single path-discriminant model. In model-discriminant HMMs, one model is constructed for each different word

[32], [15], [75], [4] while in the path discriminant HMM only one global model is constructed [50]. In the former case, each word model is assessed to determine which is most likely to have produced a given set of observations. In the latter case, word recognition is performed by finding the most likely paths through the unique model, each path being equivalent to a sequence of letters. Path discriminant HMMs can handle large vocabularies, but are generally less accurate than model-discriminant HMMs. They may incorporate a lexicon comparison module in order to ignore invalid letter sequences obtained by path optimization.

Calculation of the best paths in the HMM model is usually done by means of the Viterbi algorithm. Transition and observed feature probabilities can be learned using the Baum-Welch algorithm. Starting from an initial evaluation, HMM probabilities can be re-estimated using frequencies of observations measured on the training set [33].

First order Markov models are employed in most applications; in [50], an example of a second order HMM is given. Models for cursive script ordinarily assume discrete feature values. However, continuous probability densities may also be used, as in [3].

3.3.2 Non-Markov Approaches

A method stemming from concepts used in machine vision for recognition of occluded objects is reported in [16]. Here, various features and their positions of occurrence are recorded for an image. Each feature contributes an amount of evidence for the existence of one or more characters at the position of occurrence. The positions are quantized into bins such that the evidence for each character indicated in a bin can be summed to give a score for classification. These scores are subjected to contextual processing using a predefined lexicon in order to recognize words. The method is being applied to text printed in a known proportional font.

A method that recognizes word feature graphs is presented in [71]. This system attempts to match subgraphs of features with predefined character prototypes. Different alternatives are represented by a directed network whose nodes correspond to the matched subgraphs. Word recognition is performed by searching for the path that gives the best interpretation of the word features. The characters are detected in the order defined by the matching quality. These can overlap or can be broken or underlined.

This family of recognition-based approaches has more often been aimed at cursive handwriting recognition. Probabilistic relaxation was used in [37] to read off-line handwritten words. The model was working on a hierarchical description of words derived from a skeletal representation. Relaxation was performed on the nodes of a stroke graph and of a letter graph where all possible segmentations were kept. Complexity was progressively reduced by keeping only the most likely solutions. N-gram statistics were also introduced to discard illegible combinations of letters. A major drawback of this technique is that it requires intensive computation.

Tappert employed Elastic Matching to match the drawing of an unknown cursive word with the possible sequences of letter prototypes [80]. As it was an on-line method, the unknown word was represented by means of the angles and y-location of the strokes joining digitization points. Matching was considered as a path optimization problem in a lattice where the sum of distance between these word features and the sequences of letter prototypes had to be minimized. Dynamic programming was used with a warping function that permitted the process to skip unnecessary features. Digram statistics and segmentation constraints were eventually added to improve performance.

Several authors proposed a Hypothesis Testing and Verification scheme to recognize handprinted [44] or on-line cursive words [5], [67]. For example, in the system proposed in [5] a sequence of structural features (like x- and y-extrema, curvature signs, cusps, crossings, penlifts, and closures) was extracted from the word to generate all the legible sequences of letters. Then, the "aspect" of the word (which was deduced from ascender and descender detection) was taken into account to chose the best solution(s) among the list of generated words. In [67], words and letters were represented by means of tree dictionaries: Possible words were described by a letter tree (also called a "trie") and letters were described by a feature tree. The letters were predicted by finding in the letter tree the paths compatible with the extracted features and were verified by checking their compatibility with the word dictionary.

Hierarchical grouping of on-line features was proposed in [39]. The words were described by means of a hierarchical description where primitive features were progressively grouped into more sophisticated representations. The first level corresponded to the "turning points" of the drawing, the second level was dealing with more sophisticated features called "primary shapes," and finally, the third level was a trellis of tentative letters and ligatures. Ambiguities were resolved by contextual analysis using letter quadgrams to reduce the number of possible words and a dictionary lookup to select the valid solution(s).

A different approach uses the concept of regularities and singularities [77]. In this system, a stroke graph representing the word is obtained after skeletonization. The "singular parts," which are supposed to convey most of the information, were deduced by eliminating "regular part" of the word (the sinusoid-like path joining all cursive ligatures). The most robust features and characters (the "anchors") were then detected from a description chain derived from these singular parts and dynamic matching was used for analyzing the remaining parts.

A top-down directed word verification method called "backward matching" (see Fig. 14) is proposed in [54]. In cursive word recognition, all letters do not have the same discriminating power, and some of them are easier to recognize. So, in this method, recognition is not performed in a left-to-right scan, but follows a "meaningful" order which depends on the visual and lexical significance of the letters. Moreover, this order also follows an edge-toward-center movement, as in human vision [82]. Matching between symbolic and physical descriptions can be performed at the letter, feature and even sub-feature levels. As the system knows in advance what it is searching for, it can make use of high-level contextual knowledge to improve recognition, even at low-level stages. This system is an attempt to pro-

vide a general framework allowing efficient cooperation between low-level and high-level recognition processes.

4 Mixed Strategies: "Oversegmenting"

Two radically different segmentation strategies have been considered to this point. One (Section 2) attempts to choose the correct segmentation points (at least for generating graphemes) by a general analysis of image features. The other strategy (Section 3) is at the opposite extreme. No dissection is carried out. Classification algorithms simply do a form of model-matching against image contents.

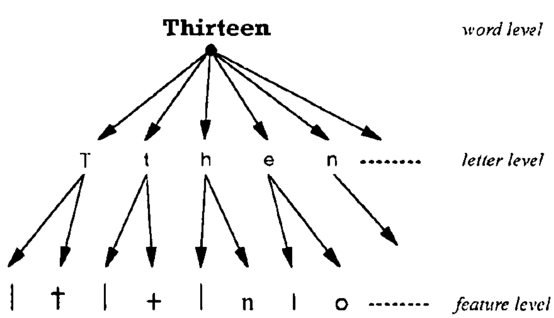


Fig. 14. Backward matching. Recognition is performed by matching an image against various candidate words, the most distinctive letters being matched first. In this example, the algorithm first seeks the letter "T" in the image, then "t," "h," and so on. These letters are more informative visually because they contain ascenders, and lexically because they are consonants.

In this section, intermediate approaches, essentially hybrids of the first two, are discussed. This family of methods also uses presegmenting, with requirements that are not as strong as in the grapheme approach. A dissection algorithm is applied to the image, but the intent is to "oversegment," i.e., to cut the image in sufficiently many places that the correct segmentation boundaries are included among the cuts made, as in Fig. 15. Once this is assured, the optimal segmentation is defined by a subset of the cuts made. Each subset implies a segmentation hypothesis, and classification is brought to bear to evaluate the different hypotheses and choose the most promising segmentation.

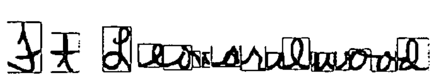


Fig. 15. Oversegmenting. Note that letters that contain valleys have been dissected into multiple parts. However, no merged characters remain, so that a correct segmentation can be produced by recombination of some of the segments.

The strategy in a simple form is illustrated in [29]. Here, a great deal of effort was expended in analyzing the shapes of pairs of touching digits in the neighborhood of contact, leading to algorithms for determining likely separation boundaries. However, multiple separating points were tested, i.e., the touching character pair was oversegmented. Each candidate segmentation was tested separately by classification, and the split giving the highest recognition confidence was accepted. This approach reduced segmentation errors 100-fold compared with the previously used segmentation technique that did not employ recognition confidence.

Because touching was assumed limited to pairs, the above method could be implemented by splitting a single image along different cutting paths. Thus, each segmentation hypothesis was generated in a single step. When the number of characters in the image to be dissected is not known a priori, or if there are many touching characters, e.g., cursive writing, then it is usual to generate the various hypotheses in two steps. In the first step, a set of likely cutting paths is determined, and the input image is divided into elementary components by separating along each path. In the second step, segmentation hypotheses are generated by forming combinations of the components. All combinations meeting certain acceptability constraints (such as size, position, etc.) are produced and scored by classification confidence. An optimization algorithm, typically implemented on dynamic programming principles and possibly making use of contextual knowledge, does the actual selection.

A number of researchers began using this basic approach at about the same time, e.g., [46], [9], [13]. Lexical matching is included in the overall process in [26] and [78].

It is also possible to carry out an oversegmenting procedure sequentially by evaluating trial separation boundaries [2]. In this work, a neural net was trained to detect likely cutting columns for machine printed characters using neighborhood characteristics. Using these as a base, the optimization algorithm recursively explored a tree of possible segmentation hypotheses. The left column was fixed at each step, and various right columns were evaluated using recognition confidence. Recursion is used to vary the left column as well, but pruning rules are employed to avoid testing all possible combinations.

5 Holistic Strategies

A holistic process recognizes an entire word as a unit. A major drawback of this class of methods is that their use is usually restricted to a predefined lexicon: Since they do not deal directly with letters but only with words, recognition is necessarily constrained to a specific lexicon of words. This point is especially critical when training on word samples is required: A training stage is thus mandatory to expand or modify the lexicon of possible words. This property makes this kind of method more suitable for applications where the lexicon is statically defined (and not likely to change), like check recognition. They can also be used for on-line recognition on a personal computer (or notepad), the recognition algorithm being then tuned to the writing of a specific user as well as to the particular vocabulary concerned.

Whole word recognition was introduced by Earnest at the beginning of the 1960s [21]. Although it was designed for on-line recognition, his method followed an off-line methodology: Data was gathered by means of a "photo-style" in a binary matrix and no temporal information was

used. Recognition was based on the comparison of a collection of simple features extracted from the whole word against a lexicon of "codes" representing the "theoretical" shape of the possible words. Feature extraction was based on the determination of the *middle zone* of the words and ascenders and descenders were found by considering the part of the writing exceeding this zone. The lexicon of possible word codes was obtained by means of a transcoding table describing all the usual ways of writing letters.

This strategy still typifies recent holistic methods. Systems still use middle zone determination to detect the ascenders and descenders. The type of extracted features also remains globally the same (ascenders, descenders, directional strokes, cusps, diacritical marks, etc.). Finally, holistic methods, as illustrated in Fig. 16, usually follow a two-step scheme:

- The first step performs feature extraction.
- The second step performs global recognition by comparing the representation of the unknown word with those of the references stored in the lexicon.

Thus, conceptually, holistic methods use the "classical approach" defined in Section 1, with complete words as the symbols to be recognized. The main advances in recent techniques reside in the way comparison between hypotheses and references is performed. Recent comparison techniques are more flexible and better take into account the dramatic variability of handwriting. These techniques (which were originally introduced for speech recognition) are generally based on Dynamic Programming with optimization criteria based either on distance measurements or on a probabilistic framework. The first type of method is based on Edit Distance, DP-matching or similar algorithms, while the second one uses Markov or Hidden Markov Chains.

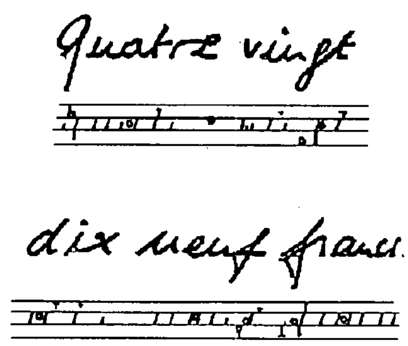


Fig. 16. Holistic recognition. Recognition consists of comparing a lexicon of word descriptions against a sequence of features obtained from an unsegmented word image. The detected features, shown below the images, include loops, oriented strokes, ascenders and descenders.

Dynamic Programming was employed in [62] and [70] for check and city name recognition. Words were represented by a list of features indicating the presence of ascenders, descenders, directional strokes and closed loops. The "middle zone" was not delimited by straight lines, but by means of smooth curves following the central part of the word, even if slanted or irregular in size. Relative y-location was associated to every feature and uncertainty coefficients were introduced to make this representation more tolerant to distortion by avoiding binary decisions. A similar scheme was used in [68] and [56], but with different features. In the first case, features were based on the notion of "guiding points," (which are the intersection of the letters and the median line of the word), whereas in the latter case they are derived from the contours of words.

One of the first systems using Markov Models was developed by Farag in 1979 [25]. In this method, each word is seen as a sequence of oriented strokes which are coded using the Freeman code. The model of representation is a non stationary Markov Chain of the first or second order. Each word of the lexicon is represented as a list of stochastic transition matrixes and each matrix contains the transition probabilities from the *j*th stroke to the following one. The recognized word is the reference Wi of the lexicon which maximizes the joint probability P(Z, Wi) where Z is the unknown word.

Hidden Markov Models are used in [63] for the recognition of literal digits and in [33] for off-line cheque recognition. Angular representation is used in the first system to represent the feature, while structural off-line primitives are used in the second case. Moreover, this second system also implement several Markov models at different recognition stages (word recognition and cheque amount recognition). Context is taken into account via prior probabilities of words and word trigrams.

Another method for the recognition of noisy images of isolated words such as in checks was recently proposed in [35]. In the learning stage, lines are extracted from binary images of words and accumulated in prototypes called "holographs." During the test phase, correlation is used to obtain a distance between an unknown word and each word prototype. Using these distances, each candidate word is represented in the prototype space. Each class is approximated with a Gaussian density inside this space and these densities are used to calculate the probability that the word belongs to each class. Other simple holistic features (ascenders and descenders, loops, length of the word) are also used in combination with this main method.

In the machine-printed text area characters are regular so that feature representations are stable, and in a long document repetitions of the most common words occur with predictable frequency. In [45], these characteristics were combined to cluster the ten most common short words with good accuracy, as a precursor to word recognition. It was suggested that identification of the clusters could be done on the basis of unigram and bigram frequencies.

More general applications require a dynamic generation stage of holistic descriptions. This stage converts words from ASCII form to the holistic representation required by the recognition algorithm. Word representation is generated from generic information about letter and ligature representations using a reconstruction model. Word reconstruction is required by applications dealing with a dynamically defined lexicon, for instance the postal application [60] where the list of possible city names is derived

from zip code recognition. Another interesting characteristic of this last technique is that it is not used to find "the best solution" but to filter the lexicon by reducing its size (a different technique being then used to complete recognition). The system was able to achieve 50% size reduction with under 2% error.

6 Concluding Remarks

Methods for treating the problem of segmentation in character recognition have developed remarkably in the last decade. A variety of techniques has emerged, influenced by developments in related fields such as speech and online recognition. In this paper, we have proposed an organization of these methods under three basic strategies, with hybrid approaches also identified. It is hoped that this comprehensive discussion will provide insight into the concepts involved, and perhaps provoke further advances in the area.

The difficulty of performing accurate segmentation is determined by the nature of the material to be read and by its quality. Generally, missegmentation rates for unconstrained material increase progressively from machine print to handprint to cursive writing. Thus, simple techniques based on white separations between characters suffice for clean fixed-pitch typewriting. For cursive script from many writers and a large vocabulary, at the other extreme, methods of ever increasing sophistication are being pursued. Current research employs models not only of characters, but also words and phrases, and even entire documents, and powerful tools such as HMM, neural nets, contextual methods are being brought to bear. While we have focused on the segmentation problem it is clear that segmentation and classification have to be treated in an integrated manner to obtain high reliability in complex cases.

The paper has concentrated on an appreciation of principles and methods. We have not attempted to compare the effectiveness of algorithms, or to discuss the crucial topic of evaluation. In truth, it would be very difficult to assess techniques separate from the systems for which they were developed. We believe that wise use of context and classifier confidence has led to improved accuracies, but there is little experimental data to permit an estimation of the amount of improvement to be ascribed to advanced techniques. Perhaps with the wider availability of standard databases, experimentation will be carried out to shed light on this issue.

We have included a list of references sufficient to provide a more-detailed understanding of the approaches described. We apologize to researchers whose important contributions may have been overlooked.

Acknowledgments

An earlier, abbreviated version of this survey was presented at ICDAR 95 in Montreal, Canada. Prof. George Nagy and Dr. Jianchang Mao read early drafts of the paper and offered critical commentaries that have been of great use in the revision process. However, to the authors falls full responsibility for faults of omission or commission that remain.

Richard G. Casey's research for this paper was performed during his sabbatical at École Nationale Supérieure des Télécommunications in Paris.

References

[1] H.S. Baird, S. Kahan, and T. Pavlidis, "Components of an Omnifont Page Reader," *Proc. Eighth Int'l Conf. Pattern Recognition*, Paris, pp. 344-348, 1986.

[2] T. Bayer, U. Kressel, and M. Hammelsbeck, "Segmenting Merged Characters," *Proc. 11th Int'l Conf. Pattern Recognition*, vol. 2. conf. B: Pattern Recognition, Methodology, and Systems, pp. 346-349, 1992.

[3] E.J. Bellegarda, J.R. Bellegarda, D. Nahamoo, and K.S. Nathan, "A Probabilistic Framework for On-line Handwriting Recognition," *Pre-Proc. IWFHR III*, Buffalo, N.Y., p. 225, May 1993.

[4] S. Bercu and G. Lorette, "On-line Handwritten Word Recognition: An Approach Based on Hidden Markov Models," *Pre-Proc. IWFHR III*, Buffalo, N.Y., p. 385, May 1993.

[5] M. Berthod and S. Ahyan, "On Line Cursive Script Recognition: A Structural Approach with Learning," *Proc. Fifth Int'l Conf. Pattern Recognition*, p. 723, 1980.

[6] M. Bokser, "Omnidocument Technologies," *Proc. IEEE*, vol. 80, no. 7, pp. 1,066-1,078, July 1992.

[7] R. Bozinovic and S.N. Srihari, "String Correction Algorithm for Cursive Script Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, no. 6, pp. 655-663, June 1982.

[8] R.M. Bozinovic and S.N. Srihari, "Off-Line Cursive Script Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 1, p. 68, Jan. 1989.

[9] T. Breuel, "Design and Implementation of a System for Recognition of Handwritten Responses on US Census Forms," *Proc. IAPR Workshop Document Analysis Systems*, Kaiserlautern, Germany, Oct. 1994.

[10] C.J.C. Burges, J.I. Be, and C.R. Nohl, "Recognition of Handwritten Cursive Postal Words using Neural Networks," *Proc. USPS Fifth Advanced Technology Conf.*, p. A-117, Nov./Dec. 1992.

[11] R.G. Casey and G. Nagy, "Recursive Segmentation and Classification of Composite Patterns," *Proc. Sixth Int'l Conf. Pattern Recognition*, p. 1,023, 1982.

[12] R.G. Casey, "Text OCR by Solving a Cryptogram," *Proc. Eighth Int'l Conf. Pattern Recognition*, Paris, pp. 349-351, Oct. 1986.

[13] R.G. Casey, "Segmentation of Touching Characters in Postal Addresses," *Proc. Fifth US Postal Service Technology Conf.*, Washington D.C., 1992.

[14] M. Cesar and R. Shinghal, "Algorithm for Segmenting Handwritten Postal Codes," *Int'l J. Man Machine Studies*, vol. 33, no. 1, pp. 63-80, July 1990.

[15] M.Y. Chen and A. Kundu, "An Alternative to Variable Duration HMM in Handwritten Word Recognition," *Pre-Proc. IWFHR III*, Buffalo, N.Y., p. 82, May 1993.

[16] C. Chen and J. DeCurtins, "Word Recognition in a Segmentation-Free Approach to OCR," *Proc. Int'l Conf. Document Analysis and Recognition*, Tsukuba City, Japan, pp. 573-576, Oct. 1993.

[17] M. Cheriet, Y.S. Huang, and C.Y. Suen, "Background Region-Based Algorithm for the Segmentation of Connected Digits," *Proc. 11th Int'l Conf. Pattern Recognition*, vol. 2, p. 619, Sept. 1992.

[18] M. Cheriet, "Reading Cursive Script by Parts," *Pre-Proc. IWFHR III*, Buffalo, N.Y., p. 403, May 1993.

[19] G. Dimauro, S. Impedovo, and G. Pirlo, "From Character to Cursive Script Recognition: Future Trends in Scientific Research," *Proc. 11th Int'l Conf. Pattern Recognition*, vol. 2, p. 516, Aug. 1992.

[20] C.E. Dunn and P.S.P. Wang, "Character Segmenting Techniques for Handwritten Text—A Survey," *Proc. 11th Int'l Conf. Pattern Recognition*, vol. 2, p. 577, Aug. 1992.

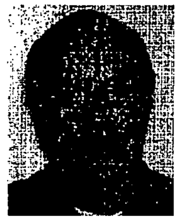
[21] L.D. Earnest, "Machine Recognition of Cursive Writing," C. Cherry, ed., *Information Processing*, pp. 462-466. London: Butterworth, 1962.

[22] R.W. Ehrich and K.J. Koehler, "Experiments in the Contextual Recognition of Cursive Script," *IEEE Trans. Computers*, vol. 24, no. 2, p. 182, Feb. 1975.

[23] D.G. Elliman and I.T. Lancaster, "A Review of Segmentation and Contextual Analysis Techniques for Text Recognition," *Pattern Recognition*, vol. 23, no. 3/4, pp. 337-346, 1990.

[24] R.J. Evey, "Use of a Computer to Design Character Recognition Logic," *Proc. Eastern Joint Computer Conf.*, pp. 205-211, 1959.
[25] R.F.H. Farag, "Word-Level Recognition Recognition of Cursive Script," *IEEE Trans. Computers*, vol. 28, no. 2, pp. 172-175, Feb. 1979.
[26] J.T. Favata and S.N. Srihari, "Recognition of General Handwritten Words Using a Hypothesis Generation and Reduction Methodology," *Proc. Fifth USPS Advanced Technology Conf.*, p. 237, Nov./Dec. 1992.
[27] R. Fenrich, "Segmenting of Automatically Located Handwritten Numeric Strings," *From Pixels to Features III*, S. Impedovo and J.C. Simon, eds., Chapter 1, p. 47, Elsevier, 1992.
[28] P.D. Friday and C.G. Leedham, "A Pre-Segmenter for Separating Characters in Unconstrained Hand-Printed Text," *Proc. Int'l Conf. Image Proc.*, Singapore, Sept. 1989.
[29] H. Fujisawa, Y. Nakano, and K. Kurino, "Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis," *Proc. IEEE*, vol. 80, no. 7, pp. 1,079-1,092, July 1992.
[30] K. Fukushima and T. Imagawa, "Recognition and Segmentation of Connected Characters with Selective Attention," *Neural Networks*, vol. 6, no. 1, pp. 33-41, 1993.
[31] P. Gader, M. Magdi, and J-H. Chiang, "Segmentation-Based Handwritten Word Recognition," *Proc. USPS Fifth Advanced Technology Conf.*, Nov./Dec. 1992.
[32] A.M. Gillies, "Cursive Word Recognition Using Hidden Markov Models," *Proc. USPS Fifth Advanced Technology Conf.*, Nov./Dec. 1992.
[33] M. Gilloux, J.M. Bertille, and M. Leroux, "Recognition of Handwritten Words in a Limited Dynamic Vocabulary," *Pre-Proc. IWFHR III*, Buffalo, N.Y., p. 417, 1993.
[34] M. Gilloux, "Hidden Markov Models in Handwriting Recognition," *Fundamentals in Handwriting Recognition*, S. Impedovo, ed., NATO ASI Series F: Computer and Systems Sciences, vol. 124, Springer Verlag, 1994.
[35] N. Gorsky, "Off-line Recognition of Bad Quality Handwritten Words Using Prototypes," *Fundamentals in Handwriting Recognition*, S. Impedovo, ed., NATO ASI Series F: Computer and Systems Sciences, vol. 124, Springer Verlag, 1994.
[36] L.D. Harmon, "Automatic Recognition of Print and Script," *Proc. IEEE*, vol. 60, no. 10, pp. 1,165-1,177, Oct. 72.
[37] K.C. Hayes, "Reading Handwritten Words Using Hierarchical Relaxation," *Computer Graphics and Image Processing*, vol. 14, pp. 344-364, 1980.
[38] R.B. Hennis, "The IBM 1975 Optical Page Reader: System Design," *IBM J. Research and Development*, pp. 346-353, Sept. 1968.
[39] C.A. Higgins and R. Whitrow, "On-Line Cursive Script Recognition," *Proc. Int'l Conf. Human-Computer Interaction—INTERACT '84*, Elsevier, 1985.
[40] W.H. Highleyman, "Data for Character Recognition Studies," *IEEE Trans. Electrical Computation*, pp. 135-136, Mar. 1963.
[41] T.K. Ho, J.J. Hull, and S.N. Srihari, "A Word Shape Analysis Approach to Recognition of Degraded Word Images," *Pattern Recognition Letters*, no. 13, p. 821, 1992.
[42] M. Holt, M. Beglou, and S. Datta, "Slant-Independent Letter Segmentation for Off-line Cursive Script Recognition," *From Pixels to Features III*, S. Impedovo and J.C. Simon, eds., p. 41, Elsevier, 1992.
[43] R.L. Hoffman and J.W. McCullough, "Segmentation Methods for Recognition of Machine-Printed Characters," *IBM J. Research and Development*, pp. 153-65, Mar. 1971.
[44] J.J. Hull and S.N. Srihari, "A Computational Approach to Visual Word Recognition: Hypothesis Generation and Testing," *Proc. Computer Vision and Pattern Recognition*, pp. 156-161, June 1986.
[45] J. Hull, S. Khoubyari, and T.K. Ho, "Word Image Matching as a Technique for Degraded Text Recognition," *Proc. 11th Int'l Conf. Pattern Recognition*, vol. 2, conf. B, pp. 665-668, Sept. 1992.
[46] F. Kimura, S. Tsuruoka, M. Shridhar, and Z. Chen, "Context-Directed Handwritten Word Recognition for Postal Service Applications," *Proc. Fifth US Postal Service Technology Conf.*, Washington, D.C., 1992.
[47] F. Kimura, M. Shridhar, and N. Narasimhamurthi, "Lexicon Directed Segmentation-Recognition Procedure for Unconstrained Handwritten Words," *Pre-Proc. IWFHR III*, Buffalo, N.Y., p. 122, May 1993.
[48] V.A. Kovalevsky, *Character Readers and Pattern Recognition*. Washington, D.C.: Spartan Books, 1968.
[49] F. Kuhl, "Classification and Recognition of Hand-Printed Characters," *IEE Nat'l Convention Record*, pp. 75-93, Mar. 1963.
[50] A. Kundu, Y. He, and P. Bahl, "Recognition of Handwritten Words: First and Second Order Hidden Markov Model Based Approach," *Pattern Recognition*, vol. 22, no. 3, p. 283, 1989.
[51] E. Lecolinet and J-V. Moreau, "A New System for Automatic Segmentation and Recognition of Unconstrained Zip Codes," *Proc. Sixth Scandinavian Conf. Image Analysis*, Oulu, Finland, p. 585, June 1989.
[52] E. Lecolinet, "Segmentation d'images de mots manuscrits," PhD thesis, Université Pierre et Marie Curie, Paris, Mar. 1990.
[53] E. Lecolinet and J-P. Crettez, "A Grapheme-Based Segmentation Technique for Cursive Script Recognition," *Proc. Int'l Conf. Document Analysis and Recognition*, Saint Malo, France, p. 740, Sept. 1991.
[54] E. Lecolinet, "A New Model for Context-Driven Word Recognition," *Proc. Symp. Document Analysis and Information Retrieval*, Las Vegas, p. 135, Apr. 1993.
[55] E. Lecolinet and O. Baret, "Cursive Word Recognition: Methods and Strategies," *Fundamentals in Handwriting Recognition*, S. Impedovo, ed., NATO ASI Series F: Computer and Systems Sciences, vol. 124, pp. 235-263, Springer Verlag, 1994.
[56] M. Leroux, J-C. Salome, and J. Badard, "Recognition of Cursive Script Words in a Small Lexicon," *Int'l Conf. Document Analysis and Recognition*, Saint Malo, France, p. 774, Sept. 1991.
[57] S. Liang, M. Ahmadi, and M. Shridhar, "Segmentation of Touching Characters in Printed Document Recognition," *Proc. Int'l Conf. Document Analysis and Recognition*, Tsukuba City, Japan, pp. 569-572, Oct. 1993.
[58] G. Lorette and Y. Lecourtier, "Is Recognition and Interpretation of Handwritten Text: A Scene Analysis Problem?" *Pre-Proc. IWFHR III*, p. 184, Buffalo, N.Y., May 1993.
[59] Y. Lu, "On the Segmentation of Touching Characters," *Int'l Conf. Document Analysis and Recognition*, Tsukuba, Japan, pp. 440-443, Oct. 1993.
[60] S. Madhvanath and V. Govindaraju, "Holisitic Lexicon Reduction," *Pre-Proc. IWFHR III*, Buffalo, N.Y., p. 71, May 1993.
[61] M. Maier, "Separating Characters in Scripted Documents," *Proc. Eighth Int'l Conf. Pattern Recognition*, Paris, p. 1,056, 1986.
[62] J-V. Moreau, B. Plessis, O. Bourgeois, and J-L. Plagnaud, "A Postal Check Reading System," *Int'l Conf. Document Analysis and Recognition*, Saint Malo, France, p. 758, Sept. 1991.
[63] R. Nag, K.H. Wong, and F. Fallside, "Script Recognition Using Hidden Markov Models," *IEEE ICASSP*, Tokyo, pp. 2,071-2,074, 1986.
[64] T. Nartker, *ISRI 1992 Annual Report*, Univ. of Nevada, Las Vegas, 1992.
[65] T. Nartker, *ISRI 1993 Annual Report*, Univ. of Nevada, Las Vegas, 1993.
[66] K. Ohta, I. Kaneko, Y. Itamoto, and Y. Nishijima, *Character Segmentation of Address Reading/Letter Sorting Machine for the Ministry of Posts and Telecommunications of Japan*, NEC Research and Development, vol. 34, no. 2, pp. 248-256, Apr. 1993.
[67] H. Ouladj, G. Lorette, E. Petit, J. Lemoine, and M. Gaudaire, "From Primitives to Letters: A Structural Method to Automatic Cursive Handwriting Recognition," *Proc. Sixth Scandinavian Conf. Image Analysis*, Finland, p. 593, June 1989.
[68] T. Paquet and Y. Lecourtier, "Handwriting Recognition: Application on Bank Cheques," *Proc. Int'l Conf. Document Analysis and Recognition*, Saint Malo, France, p. 749, Sept. 1991.
[69] S.K. Parui, B.B. Chaudhuri, and D.D. Majumder, "A Procedure for Recognition of Connected Handwritten Numerals," *Int'l J. Systems Science*, vol, 13, no. 9, pp. 1,019-1,029, 1982.
[70] B. Plessis, A. Sicsu, L. Heute, E. Lecolinet, O. Debon, and J-V. Moreau, "A Multi-Classifier Strategy for the Recognition of Handwritten Cursive Words," *Proc. Int'l Conf. Document Analysis and Recognition*, Tsukuba City, Japan, pp. 642-645, Oct. 1993.
[71] J. Rocha and T. Pavlidis, "New Method for Word Recognition Without Segmentation," *Proc. SPIE Character Recognition Technologies*, vol. 1,906, pp. 74-80, 1993.
[72] K.M. Sayre, "Machine Recognition of Handwrittten Words: A Project Report," *Pattern Recognition*, vol. 5, pp. 213-228, 1973.
[73] J. Schuermann, "Reading Machines," *Proc. Sixth Int'l Conf. Pattern Recognition*, Munich, Germany, 1982.
[74] G. Seni and E. Cohen, "External Word Segmentation of Off-Line Handwritten Text Lines," *Pattern Recognition*, vol. 27, no. 1, pp. 41-52, Jan. 1994.

[75] A.W. Senior and F. Fallside, "An Off-line Cursive Script Recognition System Using Recurrent Error Propagation Networks," *Pre-Proc. IWFHR III*, Buffalo, N.Y., p. 132, May 1993.
[76] M. Shridhar and A. Badreldin, "Recognition of Isolated and Simply Connected Handwritten Numerals," *Pattern Recognition*, vol. 19, no. 1, p. 1, 1986.
[77] J.C. Simon, "Off-line Cursive Word Recognition," *Proc. IEEE*, p. 1,150, July 1992.
[78] R.M.K Sinha, B. Prasada, G. Houle, and M. Sabourin, "Hybrid Recognition with String Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 915-925, Sept. 1993.
[79] M.E. Stevens, "Automatic Character Recognition—A State of the Art Report," MES National Bureau of Standards Technical Note no. 112, 1961.
[80] C.C. Tappert, "Cursive Script Recognition by Elastic Matching," *IBM J. Research Development*, vol. 26, pp. 765-771, Nov. 1982.
[81] C.C. Tappert, C.Y. Suen, and T. Wakahara, "The State of the Art in On-line Handwriting Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, p. 787, Aug. 1990.
[82] I. Taylor and M. Taylor, *The Psychology of Reading*. Academic Press, 1983.
[83] S. Tsujimoto and H. Asada, "Major Components of a Complete Text Reading System," *Proc. IEEE*, vol. 80, no. 7, pp. 1,133-1,149, July 1992.
[84] J. Wang and J. Jean, "Segmentation of Merged Characters by Neural Networks and Shortest Path," *Pattern Recognition*, vol. 27, no. 5, pp. 649-658, May 1994.
[85] J.M. Westall and M.S. Narasimha, "Vertex Directed Segmentation of Handwritten Numerals," *Pattern Recognition*, vol. 26, no. 10, pp. 1,473-1,186, Oct. 1993.
[86] R.A. Wilkinson, *Proc. First Conf. Census Optical Character Recognition System*, Report No. PB92-238542/XAB, National Institute of Standards and Technology, Gaithersburg, Md., May 1992.
[87] R.A. Wilkinson, "Comparison of Massively Parallel Segmenters," National Institute of Standards and Technology technical report, Gaithersburg, Md., Sept. 1992.
[88] R.A. Wilkinson, *Proc. Second Conf. Census Optical Character Recognition Systems*, National Institute of Standards and Technology, Gaithersburg, Md., Feb. 1993.
[89] B.A. Yanikoglu and P.A. Sandon, "Recognizing Off-Line Cursive Handwriting," *Proc. Computer Vision and Pattern Recognition*, 1994.



Richard G. Casey received his Doctor of Engineering Science degree from Columbia University in 1965. He had been employed since 1963 at the IBM Research Division, Yorktown Heights, New York, while preparing his thesis on character recognition. While with IBM, he had the good fortune to work with many of the pioneers in pattern recognition. In 1968, he took a leave of absence from IBM to teach pattern recognition and other subjects at the University of Florida. When he returned to IBM in 1970, he transferred to the company's research facility (now the Almaden Research Center) in San Jose, California. In 1975, he helped organize Almaden's document recognition project; it has been continuously active since then.

Dr. Casey's research into OCR, forms data extraction, and related areas has led directly to a number of IBM products. For this research, he has been recognized with IBM awards four times. He was a member of the ISRI Advisory Board for the University of Nevada at Las Vegas in 1992-1995 and program chairman for document analysis at the 1993 symposium. He was an invited speaker at the First ICDAR, and has been a program committee member for numerous conferences on document analysis. When he retired from IBM in 1994, he went on a six-month sabbatical at École Nationale Supérieure des Télécommunications in Paris.

Eric Lecolinet received his PhD degree in computer science from Université Pierre et Marie Curie, Paris, in 1990. He worked on optical character recognition and cursive script recognition in Matra, France, from 1987 to 1990, and at the IBM Almaden Research Center, San Jose, California, from 1990 to 1992. He is now an associate professor at École Nationale Supérieure des Télécommunications in Paris. His current research interests include pattern recognition, artificial intelligence, and human-computer interaction. He is a member of the IEEE.

Whole word recognition in facsimile images

Sherkat, N. Allen, T.J.
Dept. of Comput., Nottingham Trent Univ., UK;

This paper appears in: **Document Analysis and Recognition, 1999. ICDAR Proceedings of the Fifth International Conference on**

Meeting Date: 09/20/1999 - 09/22/1999
Publication Date: 20-22 Sept. 1999
Location: Bangalore India
On page(s): 547 - 550
Reference Cited: 8
Number of Pages: xxiv+821
Inspec Accession Number: 6352887

Abstract:
This paper presents the research carried out in producing a whole recognizor f handwritten words in facsimile images. Two sets of handwritten data samples collected and converted into facsimile images. The first set comprises approxim 1600 word images from 8 writers and is used for development purposes. The consists of approximately 2000 word images from 10 writers. This set is used only. The algorithms for extraction of holistic features namely, vertical bars, h cups used in the recognizor are described. A series of test are carried out and are presented using a 200 **word** lexicon. The **holistic recognizor** produced 6 rank and 82% in top 5 alternatives. When a lexicon of 1000 words was used t reduced to 49% and 70% respectively. The future directions of the research fo improvement of recognition rate are proposed. It is envisaged that definition o features would improve the overall accuracy

Index Terms:
document image processing facsimile handwritten character **recognition** algorithms cursive handwritten **words** facsimile images holes **holistic** feature extraction **recog** testing vertical bars whole **word recognition** word images word lexicon

Documents that cite this document
There are no citing documents available in IEEE Xplore at this time.

Search Results [PDF FULL-TEXT 36 KB] PREV DOWNLOAD CITATION

Whole Word Recognition in Facsimile Images

N. Sherkat, T. J. Allen
Email: {ns,tja}@doc.ntu.ac.uk
Department of Computing, The Nottingham Trent University
Burton Street Nottingham NG1 4BU, United Kingdom

Abstract

This paper presents the research carried out in producing a whole recognizer for cursive hand written words in facsimile images. Two sets of hand written data samples are collected and converted into facsimile images. The first set comprises approximately 1600 word images from 8 writers and is used for development purposes. The second set consists of approximately 2000 word images from 10 writers. This set is used for testing only.

The algorithms for extraction of holistic features namely, vertical bars, holes and cups used in the recognizer are described. A series of test are carried out and the results are presented. Using a 200 word lexicon. The holistic recognizer produced 62% top rank and 82% in top 5 alternatives. When a lexicon of 1000 words was used these values reduced to 49% and 70% respectively. The future directions of the research for improvement of recognition rate are proposed. It is envisaged that definition of further features would improve the overall accuracy.

1. Introduction

The recognizer described in this paper forms part of a multi-level recognition engine, which provides a versatile platform for experimenting with cursive handwritten words. The engine is composed of a writing case classifier, a segmentation based recogniser and a holistic recogniser. This paper concentrates on the development of, and experiments carried out using the holistic component of the engine. The holistic recognizer is based on three landmark features namely, holes, vertical bars and cups (HVBC). The recognition algorithms are driven by a lexicon, which can be automatically constructed from an ASCII word list. The system has been designed to recognize unconstrained, lower-case Roman script from any writer. The recognizer accepts off-line (static) word image data as input. The input image is in facsimile format with a resolution of 200x100 dpi. It is assumed that all word images are bi-level. In order to operate, the recognizer must first be supplied with a template database containing a list of words together with the corresponding sets of vertical bars, loops and cups that the words are expected to contain.

2. The holistic Recognizer

Vertical bars, loops and cups were selected for recognition purposes. This selection is based on previous research in recognition of on-line cursive handwriting [7] and poor quality OCR [8]. These features, taken together, are discriminative in the sense that if all of the required bars, loops and cups can be extracted from a word image, then that word image can usually be correctly recognized. This is also likely to be the case in systems with large template databases, where the possibility of confusion between similar words is high. Also, it would be expected that the features are relatively independent, i.e. using all three features together should provide more discriminatory information than using any single feature or pair of features. The extraction of these features is also relatively straightforward and computationally inexpensive. In addition, the features have formed part of several effective recognition systems developed by other researchers. For instance, Hull et al. [6] describe a segmentation based recognizer that uses horizontal strokes, concavities and loops as discriminatory features, while Gorsky [1] describes a holistic recognizer that employs line segments, ascenders, descenders and loops.

2.1 Vertical bar extraction

Vertical bars are identified by examining the number of black pixels per unit area in each of the three zones of a word image. The number of black pixels per unit area is referred to as the pixel density. The horizontal positions within each zone at which the pixel density

reaches a maximum value are taken as likely candidates for the position of vertical bar sections. When these bar sections have been identified in all three zones, bar sections in different zones that are close to each other are joined up to form full length bars. Bar sections are extracted from all three zones, rather than from the word image as a whole, in order to minimise the effect of word slant.

For a given word image, the vertical bar extraction process consists of the following main steps:

(1) Construct vertical projection histograms
(2) Construct pixel density histograms
(3) Locate pixel density maxima
(4) Locate physical endpoints of bars
(5) Join up bars in different zones
(6) Calculate bar metrics

2.2 Loop extraction

A number of problems can occur when trying to extract basic loops from a word image. First, non-basic loops are very common, and may be difficult to distinguish from basic loops. Non-basic loops can be caused inadvertently by, for example, t-bars intersecting with main letter strokes, or can be a deliberate mannerism of a particular writer. For example, some writers habitually write certain letters, such as 'f', 'j, 'k', 'q' or 'z', with looped ascenders or descenders, while it is also common for vertical strokes, such as those that occur in the letters 'l' and 'i', to be written with a loop. Secondly, loops in handwritten text may have small gaps in them. In this paper, loops that are entirely surrounded by black pixels is referred to as closed loops, while loops that are almost entirely surrounded by black pixels, i.e. loops with small gaps, is referred to as open loops. Open loops can make recognition problematic. For example, it may be difficult to distinguish between a loop and a cup. Also, letters written near to each other can produce unexpected open loops.

Both open and closed holes in a word image are obtained from the contours of the word image. Contours are extracted by tracing around the edges of all of the connected black components in the word image. For a given component, tracing proceeds from a given starting point, and is carried out so that the black portion of the component is always on the right-hand side of the direction of travel. As black pixels on the edge of the component are encountered, they are recorded. Tracing is complete when the start point is encountered again. The array of points obtained during the tracing process is referred to as a loop contour. Both external loop contours, obtained by tracing around the outside of a black component, and internal loop contours, obtained by tracing around a hole in a black component, are recorded.

2.3 Cup Extraction

Extracting basic cups from a word image is a difficult problem, as non-basic cups are common and may be difficult to distinguish from basic cups. The strategy used for dealing with this problem is the same as the one used for loop extraction, i.e. initially all cups are found and then a number of heuristics are applied in an attempt to eliminate the non-basic ones. Experimentation has shown that this approach is effective, except in the case of cups in ligatures. These are non-basic, as ligatures do not appear in database templates. Also, they can be difficult to distinguish from basic cups, and hence it is not straightforward to eliminate them using the heuristics mentioned above.

The cups in a word image are obtained by partitioning the points on all external contours into concave, convex and plane regions. When this has been done, a cup is assumed to correspond to a concave region, or to a concave region containing, or bracketed by, one or more plane regions. Concave, convex and plane regions are assumed to correspond to runs of concave, convex and plane points (respectively). Here, a point is assumed to be concave if it has a curvature of less than zero degrees, convex if it has a curvature of greater than zero degrees, and plane if it has a curvature of precisely zero degrees.

3. Word Template Database

In this system, the database template for a word is constructed by concatenating templates representing the word's constituent letters. The template for a letter is derived by applying the feature extraction procedures, described above, to a representative set of letter images taken from the development set. This allows the most common features that occur in each letter to be determined. Usually, all of the features that occur in greater than 80% of handwritten versions of a letter are included in the letter's template, although some judgement may be exercised in applying this rule of thumb.

Some letters, most notably 'f', 'z' and 's', can assume several common forms, or allographs, in handwritten text. Currently, various ad-hoc methods, which are described further below, are used to construct and represent the templates for these letters (see figure 1).

If a feature is included in a letter template, then its properties, such as size and position, are chosen so that they are an average of the properties of all instances of the letter in the reference data set. For instance, the relative width of a letter is calculated by averaging the relative widths of all instances of the letter in the data set. All features are assumed to start and end vertically precisely at zoning lines. For instance, all descenders representing vertical bars are assumed to extend from the

bottom line to the upper base-line. Also, the vertical bars in the letters 'f' and 'z' are marked as having optional descenders. This means that these bars can be considered as mid-zone bars or as descenders. It was necessary to adopt this representation because handwritten 'f's and 'z's commonly occur as allographs with and without descenders. The template for the letter 's' was difficult to choose, as 's's in handwritten words have several commonly occurring allographs. A representation consisting of a single loop with mid-zone bars on either side was eventually chosen, as this seemed to occur the most often in practice.

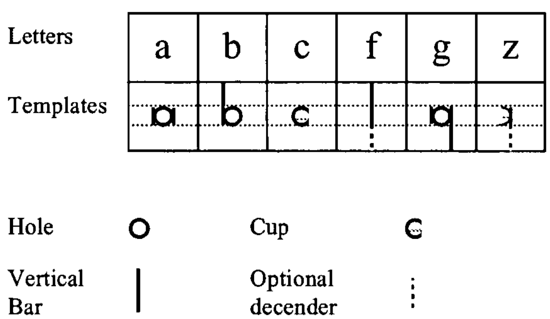


Figure 1. Sample letter templates

In this work, a feature in a word image is referred to as basic if it corresponds to a feature in the template of the word represented by the image. When constructing word templates from letter templates, a small, fixed inter-letter gap is inserted between characters. It is assumed that this gap is empty, i.e. features that might occur in ligatures, most notably cups, are not included in the template of a word. This assumption was made because, in a significant minority of cases, ligatures are omitted from handwritten words. Also, ligature features can be difficult to detect; for instance, the ligature between two characters can take the form of a straight line rather than a cup. Finally, omitting ligature features reduces the amount of processing that must be carried out during feature set matching.

4. Feature Set Matching

In order to recognize a word image, the list of features extracted from the image must be compared with all of the feature sets stored in the word template database. For each database entry, a score is required that numerically assesses the similarity between the entry and the set of features extracted from the word image. Similarity scores are calculated so that the higher the score, the greater the degree of similarity. In our system, the normalised string edit distance [2], a variant of the string edit distance [5], is used to calculate the required numerical measure of similarity.

The edit distance between the two strings of symbols A = a1a2...am and B = b1b2...bn is the minimum cost of transforming one string into the other via a sequence of elementary weighted edit operations. The operations usually allowed are

(1) substitution, whereby a symbol ai can be replaced by a symbol bj with substitution cost g(ai,bj);

(2) deletion, whereby a symbol ai can be deleted with delete cost g(ai, l) (here l represents the empty string);

(3) insertion, whereby a symbol bj can be inserted with insert cost g(l, bj).

Additional edit operations such as merges and splits and transpositions can also be used [3]. The details of cost calculation are not reproduced here due to lack of space.

5. Results and Discussion

A number of tests were carried out with the test set to investigate the recognition success and the effect of using larger word template databases. The base word set was used to construct the 200 entry database. A summary of the results is presented in Table 1.

Writer	Slant Degrees	Top (%)	Top 5 (%)
1	5 right	58.29	74.87
2	10 right	78.00	90.00
3	20 right	40.00	63.00
4	0	54.65	85.35
5	0	66.00	84.50
6	10 left	67.35	87.24
7	10 right	42.13	72.59
8	5 left	85.43	93.07
9	0	70.35	90.95
10	5 right	51.76	78.39
Overall	-	62.41	82.08

Table 1 Summary of test results for 200 words

When used with a template database containing 200 entries, the recognizer classified 62.41% of word images correctly, 82.08% of word images in the top 5 ranks and 95.57% of word in the top 50 ranks. As depicted in the table recognition results are significantly influenced by the slant of writing. This is expected since there is no provision for slanted writing during the detection of vertical bars. It is believed that a simple deslanting of the word image prior to vertical bar detection may distort the image and a more sophisticated treatment would be required. Work is currently ongoing to investigate the

possibility of detecting the slant and adjusting the principle frame of reference accordingly.

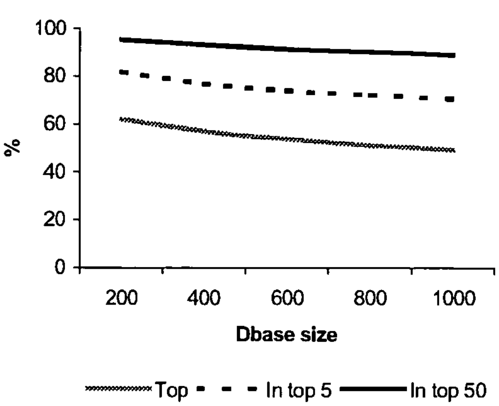
Further template databases containing 400, 600, 800 and 1000 word entries were created. These databases were constructed by selecting four additional disjoint sets of 200 words from a 15,000 word dictionary. The additional sets were randomly selected. The test results are presented in table 2.

Database size	Top (%)	In top 5 (%)	In top 50 (%)
200	62.41	82.08	95.57
400	57.17	77.15	93.46
600	53.90	74.08	91.70
800	51.38	72.42	90.59
1000	49.47	70.71	89.28

Table 2 The effect of database size on HVBC

The recognition rates deteriorate markedly with increase in the database size (figure 2). This is not far from expectation since an increase in the number of words means that there is a higher likelihood that words with similar shapes are present. As previously mentioned the additional words were selected randomly and therefore information on the effect of shape similarity is not currently available. It is interesting that the rate of deterioration seems to become asymptotic as the database size increases.

Figure 2 Recognition trend variation



These results indicate that the recognizer could be used for tasks where a relatively small number of handwritten words from different writers need to be recognized. The fact that deterioration rate decreases means that the system could also be used to reduce the size of an initial lexicon in applications where a larger number of different words have to be recognized. For example in the case of a 200 word set the search space could be reduced to 25% before an additional recogniser is engaged to further verify the results. Of course it is acknowledged that because recognition rate for the top 50 words is not quite 100% (95.57% from table 2), there is a chance that the target word could be eliminated.

Research is on going to improve the overall recognition rate of the HVBC recognizer by examining the performance of each individual feature extractor. As mentioned before various issues such as writing slant need to be addressed in order to improve the recognition results. Development of a segmentation based recognizer is also ongoing. It is planned to combine the HVBC recogniser with this recognizer. The strategy and topology of the combination forms part of the ongoing research.

6. References

[1] N. D. Gorsky (1994) Off-line recognition of bad quality handwritten words using prototypes. In S. Impedevo (Ed.), Fundamentals in Handwriting Recognition, NATO ASI Series F, 124, (pp. 199-217). Berlin: Springer-Verlag.

[2] A. Marzal and E. Vidal (1993) Computation of normalised edit distance and applications. IEEE Trans. PAMI, Vol. 15, no. 9, September 1993.

[3] G. Seni, V. Kripasundar and R.K. Srihari (1995) Generalizing edit distance for handwritten text recognition. Proceedings of SPIE - The International Society for Optical Engineering, 1995, Vol 2422, pp. 54-65, 1995. ISBN 0 8194 1769 6

[4] A. W. Senior (1994) Off-line cursive handwriting recognition using recurrent neural networks. PhD thesis, Trinity Hall, Cambridge University, Cambridge, UK, 1994.

[5] R. A. Wagner and M.J. Fischer (1974) The string-to-string correction problem. Journal ACM, Vol. 21, No. 1, Januray 1974, pp. 168-173.

[6] J. J. Hull, T.K. Ho, J. Favata, V. Govindaraju and S. Srihari (1992) Combination of segmentation- based and wholistic handwritten word recognition algorithms. S. Impedovo and J.C Simon (Eds.), From Pixels to Features III: Frontiers in Handwriting Recognition (pp. 261-272). Holland: Elsevier Science.

[7] R. K. Powalka, N. Sherkat (1995) Zoning invariant holistic recognition of handwriting, Proceedings of ICDAR95, Third international conference on Document Analysis and Recognition, pp 64-67, Montreal, Canada, August 14-16 1995

[8] G. H. Raza, N. Sherkat (1997) Recognition of facsimile documents using a database of robust features, Fourth international conference on Document Analysis and Recognition, pp 444-448, Ulm, Germany, August 18 1997

Whole word recognition in facsimile images

Sherkat, N. Allen, T.J.
Dept. of Comput., Nottingham Trent Univ., UK;

This paper appears in: **Document Analysis and Recognition, 1999. ICDAR Proceedings of the Fifth International Conference on**

Meeting Date: 09/20/1999 - 09/22/1999
Publication Date: 20-22 Sept. 1999
Location: Bangalore India
On page(s): 547 - 550
Reference Cited: 8
Number of Pages: xxiv+821
Inspec Accession Number: 6352887

Abstract:
This paper presents the research carried out in producing a whole recognizor f handwritten words in facsimile images. Two sets of handwritten data samples collected and converted into facsimile images. The first set comprises approxim 1600 word images from 8 writers and is used for development purposes. The consists of approximately 2000 word images from 10 writers. This set is used only. The algorithms for extraction of holistic features namely, vertical bars, h cups used in the recognizor are described. A series of test are carried out and are presented using a 200 **word** lexicon. The **holistic recognizor** produced 6 rank and 82% in top 5 alternatives. When a lexicon of 1000 words was used t reduced to 49% and 70% respectively. The future directions of the research fo improvement of recognition rate are proposed. It is envisaged that definition o features would improve the overall accuracy

Index Terms:
document image processing facsimile handwritten character **recognition** algorithms cursive handwritten **words** facsimile images holes **holistic** feature extraction **recog** testing vertical bars whole **word recognition** word images word lexicon

Documents that cite this document
There are no citing documents available in IEEE Xplore at this time.

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

The role of holistic paradigms in handwritten word recognition

Madhvanath, S. Govindaraju, V.
IBM Almaden Res. Center, San Jose, CA, USA;

This paper appears in: **Pattern Analysis and Machine Intelligence, IEEE Transactions on**

Publication Date: Feb. 2001
On page(s): 149 - 164
Volume: 23 , Issue: 2
ISSN: 0162-8828
Reference Cited: 56
CODEN: ITPIDJ
Inspec Accession Number: 6927556

Abstract:
The **holistic** paradigm in handwritten word **recognition** treats the word as a indivisible entity and attempts to recognize words from their overall shape, as their character contents. In this survey, we have attempted to take a fresh loo potential role of the **holistic** paradigm in handwritten word **recognition**. The begins with an overview of studies of reading which provide evidence for the e a parallel **holistic** reading process,in both developing and skilled readers. In w believe is a fresh perspective on **handwriting recognition**, approaches to **re** are characterized as forming a continuous spectrum based on the visual comp the unit of **recognition** employed and an attempt is made to interpret well-kn paradigms of word **recognition** in this framework. An overview of features, methodologies, representations, and matching techniques employed by **holist** approaches is presented

Index Terms:
handwritten character **recognition** handwritten word **recognition** **holistic** paradigms entity parallel **holistic** reading process visual complexity word shape

Documents that cite this document
There are no citing documents available in IEEE Xplore at this time.

Search Results [PDF FULL-TEXT 2408 KB] DOWNLOAD CITATION

The Role of Holistic Paradigms in Handwritten Word Recognition

Sriganesh Madhvanath, *Member, IEEE*, and Venu Govindaraju, *Senior Member, IEEE*

Abstract—The Holistic paradigm in handwritten word recognition treats the word as a single, indivisible entity and attempts to recognize words from their overall shape, as opposed to their character contents. In this survey, we have attempted to take a fresh look at the potential role of the Holistic paradigm in handwritten word recognition. The survey begins with an overview of studies of reading which provide evidence for the existence of a parallel holistic reading process in both developing and skilled readers. In what we believe is a fresh perspective on handwriting recognition, approaches to recognition are characterized as forming a continuous spectrum based on the visual complexity of the unit of recognition employed and an attempt is made to interpret well-known paradigms of word recognition in this framework. An overview of features, methodologies, representations, and matching techniques employed by holistic approaches is presented.

Index Terms—Handwriting recognition, holistic paradigms, analytical methods, reading theory, pattern recognition.

1 Introduction

Handwritten Word Recognition (HWR), also called *Isolated* Handwritten Word Recognition, deals with the problem of machine reading handwritten words. There are two different problems that fall under the purview of handwritten word recognition: Offline HWR and Online HWR.

Offline HWR deals with the problem of reading a handwritten word *offline*, that is, at some point in time (minutes, months, years) after it was written. A handwritten word is typically scanned in from a paper document and made available in the form of a binary or gray-scale image to the recognition algorithm.

The problem differs from *online* HWR where the writing is with a special pen on an electronic notepad or a tablet and where temporal information, such as the position and velocity of the pen along its trajectory, is available to the recognition algorithm. Since most algorithms for online HWR attempt to recognize the writing as it is being written, online HWR is also sometimes referred to as "real-time" HWR.

This survey focuses on the task of offline HWR. However, the discussion is pertinent to the online problem as well.

1.1 The Offline HWR Task

Some applications of offline HWR today are recognition of handwritten check amounts, interpretation of handwritten addresses on pieces of mail, reading handwritten responses on forms, and automatic filing of faxes. The handwritten text must be located, extracted, made free of artifacts stemming from the medium (underlines and background from the check leaf, boxes from forms, postal marks from the piece of mail), separated into lines if necessary, and, finally, into individual words before it can be recognized. These steps are generally nontrivial and research issues in their own right. We assume in this survey that the complex task of segmentation of the image of the handwritten word or phrase of interest from its surroundings has already been accomplished by prior processes. The tasks of segmentation and recognition of words are generally accomplished sequentially based upon different features of the image. They are consequently difficult to combine, except superficially in the sense that word recognition is used to choose from multiple word segmentation hypotheses. We will focus in this survey on the task of *recognition* of the isolated word or phrase using the appropriate lexicon (Fig. 1).

The handwritten word or phrase may be constrained by the application to be in a particular style. For example, forms often request that the responses be handprinted. In general, however, handwritten words may be cursive, purely discrete, touching discrete, or a mixture of these styles (Fig. 2). While for some applications of online HWR, a single author assumption can be made and the algorithms tuned to a particular style of writing, this assumption cannot generally be made for the offline problem. Consequently, the recognition algorithm must deal with a variety of author-specific idiosyncrasies.

Moreover, there is little or no control in most offline scenarios on the type of medium and instrument used. The artifacts of the complex interactions between medium, instrument, and subsequent operations such as scanning and binarizations present additional challenges to algorithms for offline HWR. Offline HWR is, therefore, generally regarded as much more difficult than its online counterpart.

- *S. Madhvanath is with the IBM Almaden Research Center, San Jose, CA 95120. E-mail: srig@almaden.ibm.com.*
- *V. Govindaraju is with the Center of Excellence for Document Analysis and Recognition (CEDAR), Department of Computer Science and Engineering, State University of New York at Buffalo, 226 Bell Hall Amherst, NY 14260. E-mail: govind@cedar.buffalo.edu.*

Manuscript received 18 Nov. 1999; revised 18 Apr. 2000; accepted 6 Nov. 2000.
Recommended for acceptance by R. Plamondon.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110966.

0162-8828/01/$10.00 © 2001 IEEE

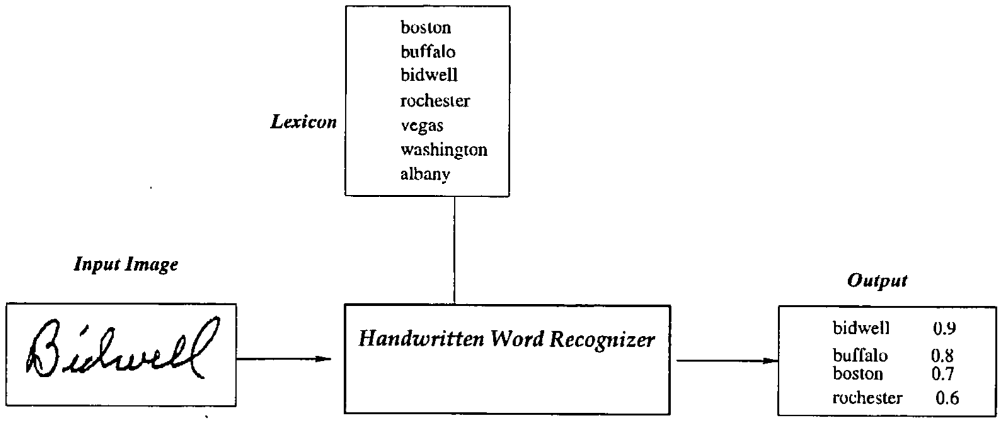


Fig. 1. I/O behavior of offline HWR. Input is the word image and a lexicon of possible choices. Output is the lexicon sorted by some confidence measure.

(a) (b) (c)

Fig. 2. Examples of handwriting styles. (a) Cursive, (b) discrete touching, and (c) mixed.

Since words are fairly complex patterns and owing to the great variability in handwriting style, the HWR task is a difficult one. In fact, it is only made tractable when a lexicon of valid words is provided. The lexicon is usually determined by the application domain. For example, there are only 33 different words that may appear in the so-called legal amounts on handwritten checks. The lexicon for this application, is hence, both small and *static*, i.e., constant across all recognition instances (Fig. 3).

The lexicon used for street name recognition in Handwritten Address Interpretation (HWAI) is generally comprised of street name candidates generated from knowledge of the zip code and the street number. This is an example of an HWR application where the lexicon is *dynamic*, i.e., varying from one instance to the next (Fig. 4). Some applications, such as the reading of handwritten prose, may involve very large lexicons of over 20,000 words. The nature of the lexicon is crucial to the design of HWR algorithms for a particular application.

1.2 Holistic Approaches

From the earliest days of research in HWR, two approaches to the problem have been identified. The first approach, often called the *analytical* approach, treats a word as a collection of simpler subunits such as characters and proceeds by segmenting the word into these units, identifying the units and building a word-level interpretation using the lexicon. The other approach treats the word as a single, indivisible entity and attempts to recognize it using features of the word as whole. The latter approach is referred to as the *word-based* or *holistic* approach and is

one	two	three	four	five
six	seven	eight	nine	ten
eleven	twelve	thirteen	fourteen	fifteen
sixteen	seventeen	eighteen	nineteen	twenty
thirty	forty	fifty	sixty	seventy
eighty	ninety	hundred	thousand	dollars
dollar	and	only		

Fig. 3. Handwritten legal amount recognition involves the recognition of each word in the phrase matched against a static lexicon of about 33 words.

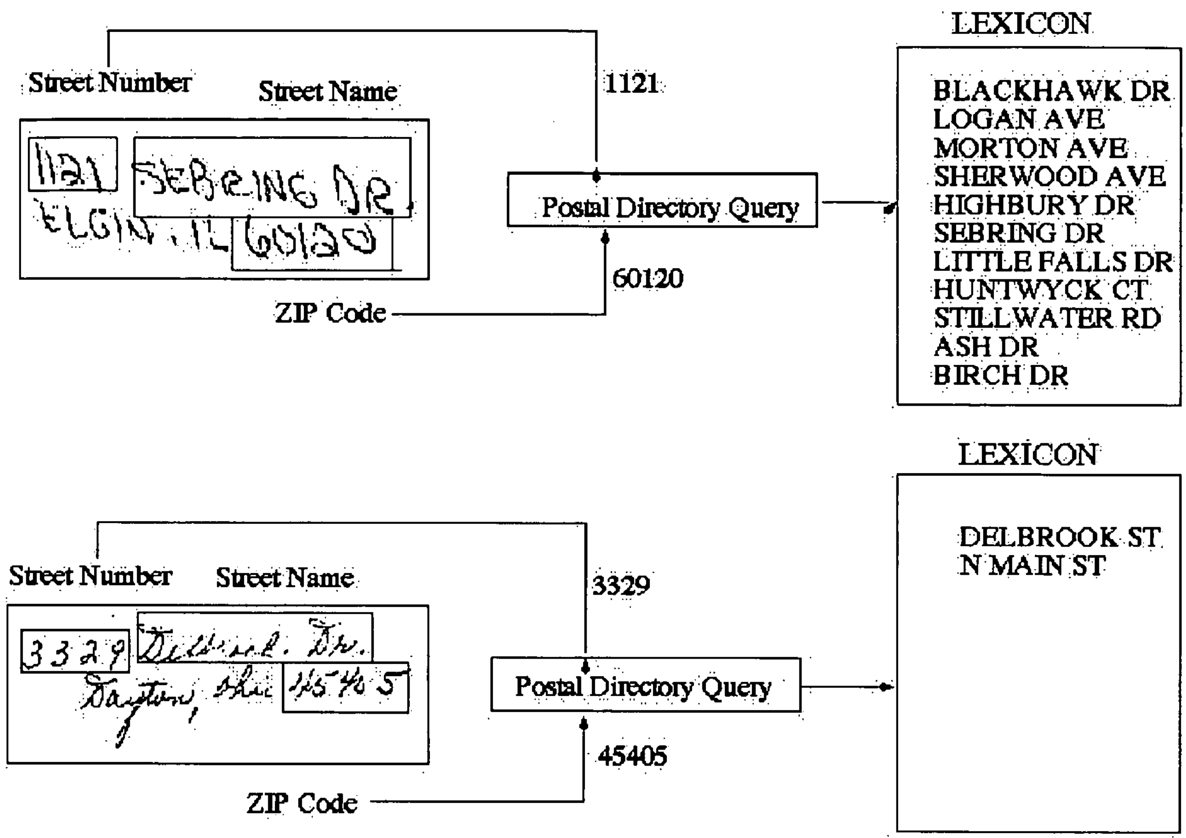


Fig. 4. A lexicon is dynamically created using the zip code and the street number. Note that the street name images are matched with different lexicons generated. (a) In zip code 60120 there are 11 streets which have the street number 1121. (b) In zip code 45405 there are two streets with the street number 3329.

inspired in part by psychological studies of human reading, which indicate that humans use features of word shape such as *length, ascenders, and descenders* in reading (Fig. 5).

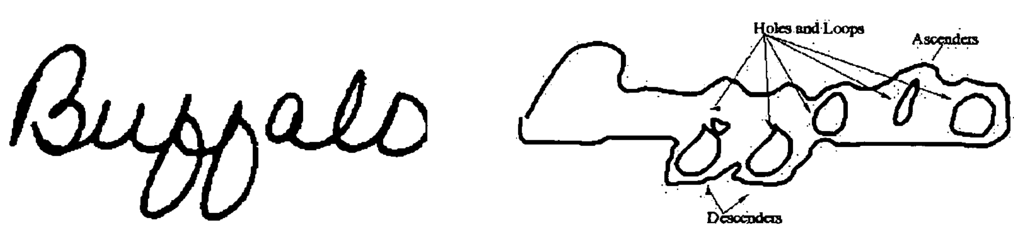
Because analytical approaches decompose HWR into the problem of identifying a sequence of smaller subunits, the chief problems they face are 1) *segmentation ambiguity*: deciding where to segment the word image (Fig. 6) and 2) *variability of segment shape*: determining the identity of each segment (Fig. 7), [13].

Holistic approaches circumvent these problems because they make no attempt to segment the word into subunits. Instead, they rely on features and matching at the word-level to determine the identity of the word.

1.3 Relevance of the Holistic Paradigm

Analytical approaches that decompose handwritten words into characters or other subunits derived from characters do not generally distinguish between static and dynamic lexicons; random strings of characters are recognized as effectively as valid words.

For holistic approaches, on the other hand, every word is a different class. The holistic features and matching scheme



(a) (b)

Fig. 5. (a) Word image. (b) Word-shape features do not refer to individual characters and include length, ascenders, descenders, loops, etc.

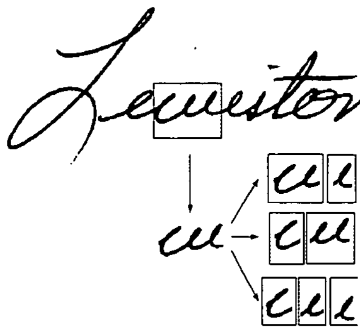


Fig. 6. Ambiguities in segmentation: The letter(s) following the "e" can be "w," or "ui" or "iu" or "iii."

used must be coarse enough to be stable across exemplars of the same class(word), i.e., across a variety of writing styles, but fine enough to be able to distinguish exemplars of different classes. Given that words are complex 2D patterns and given the large variety of writing styles, these are difficult criteria to satisfy when the number of classes is large or unknown. Hence, holistic approaches have been used traditionally in application scenarios wherein the classes are few and fixed. For example, the check amount recognition task (Fig. 3). Moreover, when the lexicon is small and static, it becomes possible to collect a large number of training samples of each class. Training may then be performed in the traditional sense of estimating class-conditional densities of features from the training samples or storing prototypical feature vector exemplars for each class.

When the lexicon is large or dynamic [11], [22], [23], [34] (handwritten address interpretation example in Fig. 4), the ability of any given set of holistic features to distinguish between word classes is diminished. In addition, it is difficult or impossible from a practical standpoint to obtain representative samples of all word classes for training a holistic classifier. For these reasons, there is consensus among researchers in the field of HWR that the utility of the holistic approach is either in the small, static lexicon scenario or in the filtering of large lexicons. For example, a survey of the state of the art in online HWR [52] concludes that

> While the [whole-word] approach can be useful for small vocabularies, current thinking is that it is not viable for the general problem [of classification of handwritten words].

There are two issues that must be emphasized in this context.

First, classification is only part of the problem of recognition of offline handwritten words. Given the difficulty of the task, practical recognition engines must employ multiple classification algorithms and complex strategies for combining classifier decisions [35]. Fig. 8 shows the role of a holistic recognizer in the complex combination of recognizers used by a handwritten address interpretation system [36].

Second, the merit of a particular paradigm is best judged by its cost/accuracy benefits, rather than by accuracy alone. An algorithm that is highly accurate at classifying words is not viable in practice if the computational cost involved is unreasonable. Conversely, an algorithm, such as the holistic recognizer, with relatively low accuracy may prove beneficial if used in conjunction with more accurate algorithms and if the additional computational burden is relatively small.

This investigation into holistic approaches is further motivated by the following observations:

- **Intrinsic advantages of the holistic paradigm.** By circumventing segmentation issues and treating each word as a class unto itself, holistic approaches have the potential to model effects that are unique to the class. For example, they can model *coarticulation effects*, i.e., the changes in the appearance of a character as a function of the shapes of neighboring characters. Fig. 5 shows the two "f"s written have different shapes depending on what precedes and follows them. Generally speaking, algorithms based on the holistic paradigm are computationally efficient.
- **Orthogonality of holistic features.** Holistic features provide information about the word that is clearly orthogonal to the knowledge of characters in it and it stands to reason that the introduction of this knowledge should improve recognition. For example, a Holistic approach may succeed when the writing is so poor that the individual characters cannot be distinguished but the overall shape of the word is preserved (Fig. 9).
- **Evidence from psychological studies.** A large body of evidence from psychological studies of reading (Section 2) points towards the use of a holistic approach in conjunction with analysis of letter identities—humans do not, in general, read words letter by letter. A computational theory of reading should include the holistic paradigm.
- **Potential benefits for HWR engines.** The recognition of unconstrained handwritten words is a challenging problem that may be addressed only when a lexicon is available. Existing recognition algorithms show a decline in both accuracy as well

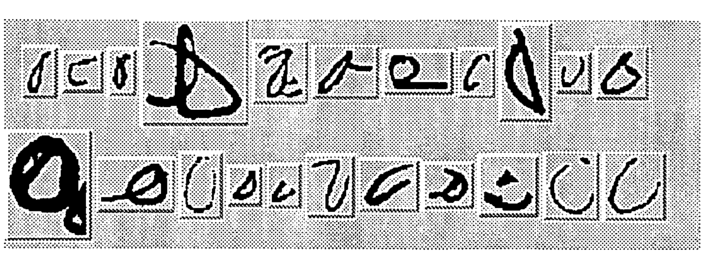


Fig. 7. Wide variability in shapes of characters ("o" in this example) even when taken from the writing of the same writer.

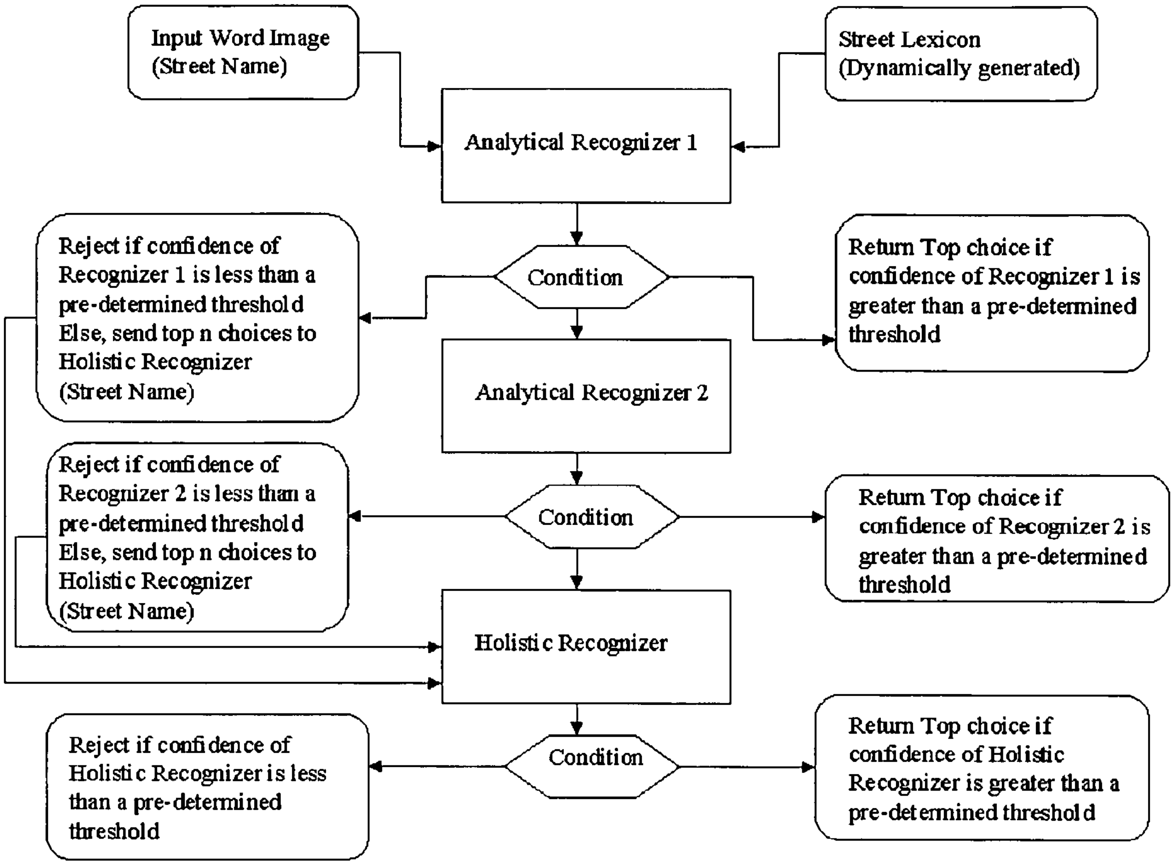


Fig. 8. Combination of three word recognizers, two analytical and a holistic in the context of handwritten address interpretation. The holistic recognizer serves as a "tie-breaker" between the top choices of the two analytical recognizers.

as computational efficiency when confronted with real-world recognition scenarios involving noisy images and large lexicons. The use of multiple classifiers with substantially different features and approaches, decision combination methods, and complex strategies for thresholding are some ways of combating the decline in performance (Fig. 8).

We hope that this survey will encourage the reader to reexamine the consensus about the rôle of the holistic paradigm in offline handwritten word recognition.

1.4 Organization of Survey

In Section 2, we discuss some of the findings from experiments with human readers. An important motivation for investigating the holistic paradigm comes from the fact that humans use holistic features in reading and tend to read whole words at a time. Psychological studies have demonstrated the robustness of human reading skills in the presence of large distortion or incomplete information at lower levels of the text hierarchy. Fluent reading appears to involve the recognition of word patterns rather than individual letter patterns. In Section 3, we attempt to refine the distinction between holistic and nonholistic approaches in order to better comprehend the methods proposed in the literature. We discuss various broad classifications of the holistic methods surveyed in Section 4 and survey holistic features, representations, and matching methodologies. The survey is summarized in the concluding section.

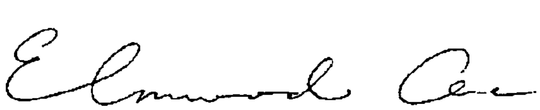


Fig. 9. Images with low character information cause problems for analytical approaches.

2 The Holistic Paradigm and the Psychology of Reading

It is no surprise that a dichotomy analogous to holistic/analytical approaches to machine recognition of words is also the center of a long-standing debate in reading studies. An excellent survey of this debate is presented by Soltysiak [51] and forms the basis for this section.

Visual recognition of words has been widely investigated by psychologists during the past century (for example, [8], [44], [56], [37], [27]) and has produced two very different interpretations. *Holistic* theories suggest that words are identified directly from their global shape; the opposing view of *hierarchical* theories is that recognition results from identifications of component letters. These theories do not hypothesize different mechanisms for

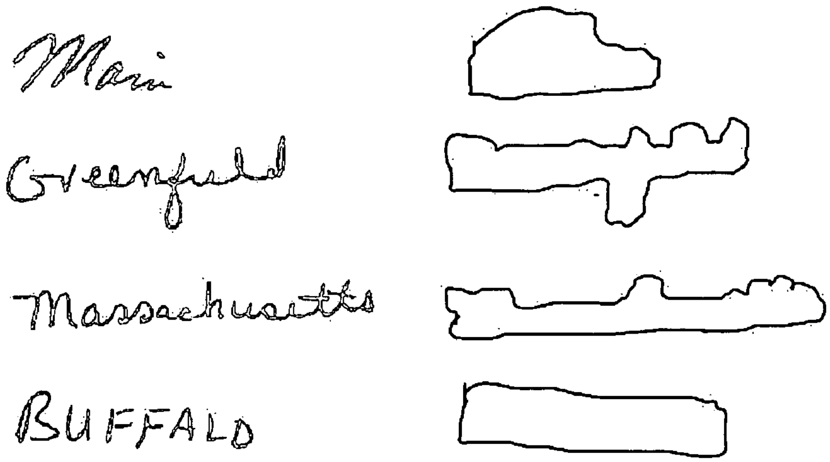


Fig. 10. The word shape of cursive words alone contains sufficient information to classify the image as one of the lexicon words. Words written completely in upper case, such as BUFFALO in the example do not posses such information.

printed and handwritten words; the studies supporting them deal primarily with printed words.

Holistic theories of reading propose that reading is accomplished using stored encodings of shapes of words. They predict that lowercase words are easier to read than uppercase words and that familiar words such as function words are easier to read than unfamiliar words. They also predict degraded recognition performance when word shape is disrupted, with this degradation being more pronounced for words compared to nonwords, and familiar words compared to unfamiliar ones.

Fig. 10 illustrates how word shape contains sufficient information to classify words in certain small lexicons. The perceptual features that are being invoked are perhaps the length of the word, the relative positions of ascenders and descenders, and other cues. It is also to be noted that if a word is written entirely in uppercase, there are no shape features present.

Hierarchical theories, on the other hand, hypothesize that words are recognized from letters and letters from features detected in the stimulus. Letter detectors are thought to contain information solely about letter identities and not their visual form and letters are thought to be processed in parallel. The role of a hierarchical mechanism in reading is widely accepted. As argued by Coltheart in 1981, abstract letter identification enables reading of words in typeface never before encountered. McClelland (1977) argued that it is identification of letters that allows words to be recognized as they are "the only invariant cues to the identity of words." There is, however, evidence to suggest that this need not be the sole means by which words are recognized. Recently, models that combine these conflicting interpretations have been proposed based on evidence from studies of individuals with acquired dyslexia (especially [25]) and studies of reading development (see, for example, [48]). These models propose that holistic and hierarchical processes operate in parallel in both the developing and the skilled reader.

Different holistic theories define word shape in different terms—word envelopes, shapes and sizes of individual letters, arrangements of ascenders, descenders and neutrals, digrams, and spelling units. Early evidence for holistic theories was provided by studies that showed that the time required to initiate naming of a word was less than that of a single letter—the well-known *word superiority effect* [8].

Moreover, word regularity effects (regular words such as MINT read aloud faster than irregular words such as PINT) and semantic priming effects (context facilitating word recognition) have been found to be more pronounced for upper than lowercase words [55] and have argued to indicate holistic recognition of words when word shape is available and more detailed analytical recognition process in the absence of such information. In another study, subjects were asked to guess the next word in a sentence given varying amounts of information about the next word [30]. It was found that guessing accuracy was enhanced when word length information was provided and further improved when word shape information was made available.

Studies involving proofreading tasks [39], [19] provide further evidence for word shape in word recognition. These tasks involved recognition of words in text passages, the words mutilated by substituting or deleting letters. Certain mutilations involved deletion of a perceptual feature such as an ascender or descender, or substitution of a perceptual feature by a neutral character, causing a large change in word shape (e.g., "fastest" became "fascest" or "fasest"). Others involved deletion of a neutral character or substitution of a neutral character by another, and caused only a small change (e.g., "fastest" became "fastect" or "fastet"). It was observed that misspellings that preserved word shape were less noticeable than those that disrupted word shape. This has been argued to support a two-stage model of visual analysis [2], [3] involving the cyclical interaction of a passive global process which selects a set of words matching the shape of the stimulus and an active local process which "fills-in" details to permit full identification of the stimulus.

The evidence for holistic recognition of words is commonly criticized as being inconclusive because of the failure of studies to independently manipulate confusion of

letters from that of word shape; apparent word shape effects may be explained at the letter level using the argument that lowercase letters are more distinct that their uppercase counterparts [21]. However, there is evidence for a parallel reading mechanism in humans that is based solely on word shape. Studies of individuals with acquired reading disorders or dyslexia suggest that different forms of dyslexia result from impairment at different levels of the human visual word recognition system. Surface dyslexia and letter-by-letter reading [48] may be explained in terms of damage to the word representation or its connections with the letter detectors.

However, there is evidence from studies of individuals with deep acquired dyslexia, especially "TM," described by Howard in [25]. TM had great difficulty matching words across case. TM was much worse at reading words with letters separated by "+" (as in "w+o+r+d+s"), while words with letters separated by spaces (w o r d s) were read as well as when not so separated. In addition, TM was significantly worse at reading case alternated (WoRdS) and diagonally written words; and unable to understand abbreviations when presented in inappropriate case (e.g., E.G.).

TM is unable to either extract or use the abstract identities of the letters that constitute the word. This directly contradicts the prediction of hierarchical theories that no word can be recognized if letter identity information is unavailable. These results are seen as evidence for a reading mechanism that is completely independent of letter identities. A visual word recognition system with two available routes has been proposed by a number of researchers. There are also a number of theories about how the two kinds of information are combined for fluent reading, which are beyond the scope of this review. It is clear from these studies, however, that word shape plays a significant role in visual word recognition both in conjunction with character identities, as well as in situations wherein component letters cannot be discerned.

This review would not be complete without some mention of the research that suggests that the mechanisms used for recognition of cursive script may differ in fundamental ways from those used for printed words [56]. Most hierarchical models of reading assume a model of parallel processing wherein features of individual letters simultaneously activate words in the mental lexicon [53]. The fact that individual letters are easy to segment from the background would suggest that word shape features such as ascenders and descenders provide little additional information to aid in the recognition of letters and, ultimately, the word. Clearly, this is not true of cursive script. In one study conducted at the Nimjen University in the Netherlands, the presence of ascenders and descenders was found to have an impact on both reading speed and error rate [47]. In particular, reading speed was seen to decrease for cursively written words which have no ascenders or descenders.

3 Paradigms for Visual Word Recognition

In the Section 1, we presented two paradigms for word recognition: analytical and holistic. In this section, we attempt to refine the distinction between these paradigms and clarify what a holistic approach is and what it is not. A review of the literature reveals a variety of interesting methods which are difficult to classify as one or the other. In addition, there appear to be at least two different senses in which the term "holistic approach" has been used in the literature: 1) an approach that matches words as a whole and 2) an approach that uses word shape features. It is important to distinguish holistic features from holistic approaches. A holistic approach may or may not use word shape features. For example, it may use pixel direction distribution features [54]. Conversely, a classifier may use word shape features in an approach that is not holistic to perform segmentation [15], [20] and/or character recognition.

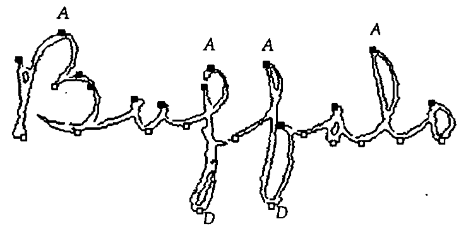


Fig. 11. Ascenders and descenders are perceptual features. Features such as the proportion of pixels in the left and right segments, number of extrema, the perimeter of the word, etc., are examples of holistic features.

The term "global features" has been used by some researchers to refer to "simpler aspects of word shape that can be easily and reliably measured" [54]. Often, this refers to estimates of word length and counts of perceptual features such as ascenders and descenders (Fig. 11).

Hierarchical theories of reading postulate the use of letter models as part of the recognition process, whereas holistic theories of reading suggest that the word identity is determined directly from word shape features extracted from the stimulus (Fig. 10). The holistic/analytical distinction differs from this holistic/hierarchical dichotomy encountered in reading studies in significant ways. Analytical approaches for handwritten word recognition are not limited to the use of letters as models; conversely, an approach that uses nonletter models would be considered analytical rather than holistic. In fact, the holistic/analytical classification is a continuous spectrum rather than a dichotomy, as is evidenced from the methods surveyed. Fig. 12 illustrates this particular point. At the one end, we have a word represented as an array of pixels and on the other as an ASCII string. The features move from the fine-grained pixels to the holistic shape of the word. In the middle, we have features ranging from the purely analytical ,such as strokes, loops, and characters, to the holistic, such as histogram profiles of words and pixel density distributions. The exact line where we depart from the analytical and move to holistic is in fact a gray band.

3.1 Features and Models

Features may be used directly to determine the identity of a word image or they may be used to determine the identity of intermediate entities which constitute the word image. We will refer to these intermediate entities as submodels. Features such as strokes of characters, loops, t-crossings,

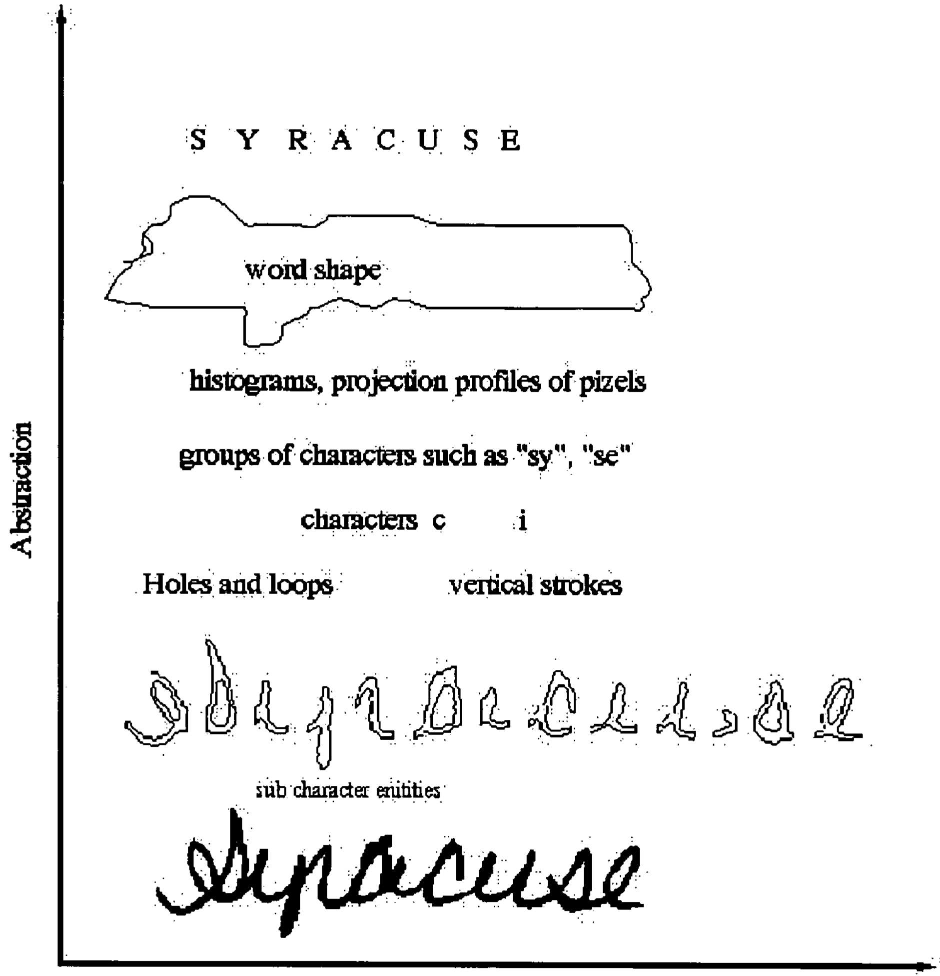


Fig. 12. The continuum of features moving from the fine-grained pixel level features to the coarser features of strokes, then characters, groups of characters and, finally, the ASCII representation of the word. The line between holistic features and analytical features is a actually subjective.

i-dots, and individual characters themselves are all regarded as submodels. The use of submodels is motivated by the realization that exemplars of different classes are not unrelated, rather, they are composed of common subentities. The burden of dealing with variations in the input may effectively be shifted to the level of submodels, where they are more constrained and easier to characterize and compensate for. The assumption implicit in the use of submodels is that they are consistent when they appear in different positions and in exemplars of different classes.

We would like to formally define a holistic approach as one that does not use submodels as part of its classification strategy. We refer to nonholistic approaches as model-based approaches. These approaches have traditionally been called "analytical," but holistic approaches may involve detailed analysis of the word as well. In fact, in an early survey of HWR [29], Frishkopf's approach of encoding the trace of the word as a sequence of extrema points and matching the entire sequence against a lexicon of similarly encoded words—essentially a holistic approach—is classified under analytical approaches. Therefore, we prefer the term "model-based" to describe approaches that employ submodels. However, we will continue to use the familiar term "analytical" to refer to such approaches.

Previous reviews of the literature in HWR have proposed similar taxonomies for methods [7]. These taxonomies have been based largely on the scheme used to segment the word image, and holistic approaches have been described as those which use no segmentation or implicit segmentation. The uniqueness of the taxonomy presented here, in our opinion, lies in the fact that it is based on submodels used in the recognition process, rather than the segmentation scheme used.

3.2 Analytical (Model-Based) Approaches

The central idea in a model-based approach is to identify parts of the word image as one of a predefined set of models known to the classifier [46]. A "circular" situation arises here: in order for a piece of the image to be identified as a model, it must first be segmented; but in order for it to be correctly segmented, it must be identified as a valid model first. Casey and Lecolinet [7] introduce the term "dissection" to refer to a partitioning of the image based on image features alone (i.e., without involving recognition). Different model-based

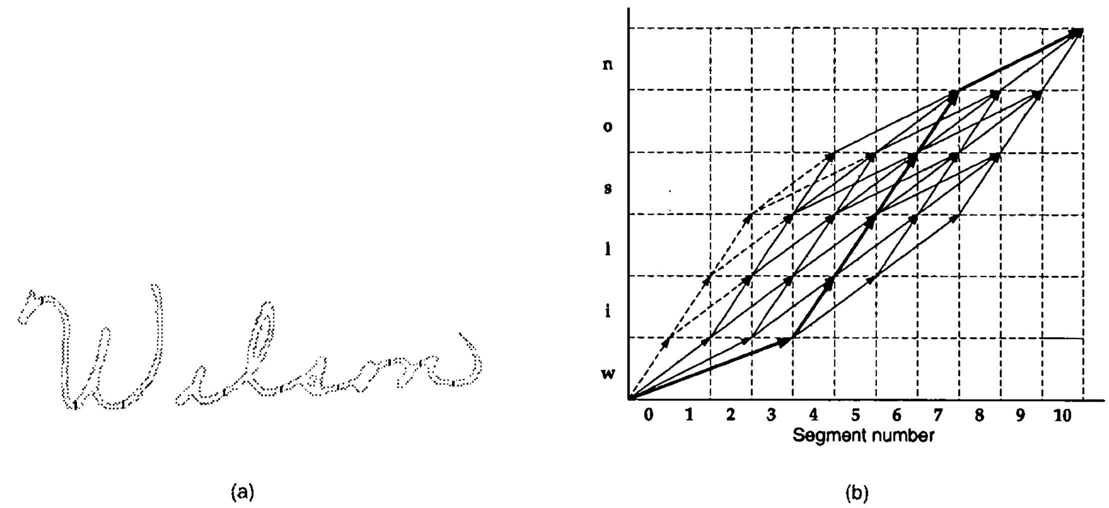


Fig. 13. (a) Image features based dissection of a word image. (b) Dynamic matching of the segments (intermediate entities) with the lexicon word WILSON.

methods place different emphasis on the subprocesses of recognition and dissection to arrive at a final segmentation such that each identified segment corresponds to one of the models. At one end of the spectrum are methods which perform an image-feature-based dissection and use recognition of segments to detect and correct segmentation errors (Fig. 13). At the other end of the spectrum are methods which scan the image looking for models. This scanning may be of either the image or some representation thereof. Here, segmentation is driven by recognition and some researchers have used the confusing term "segmentation-free" to describe such approaches. Fig. 14 illustrates this process. The recognition of different characters "peaks" as a window (of the size of a character) slides along the image.

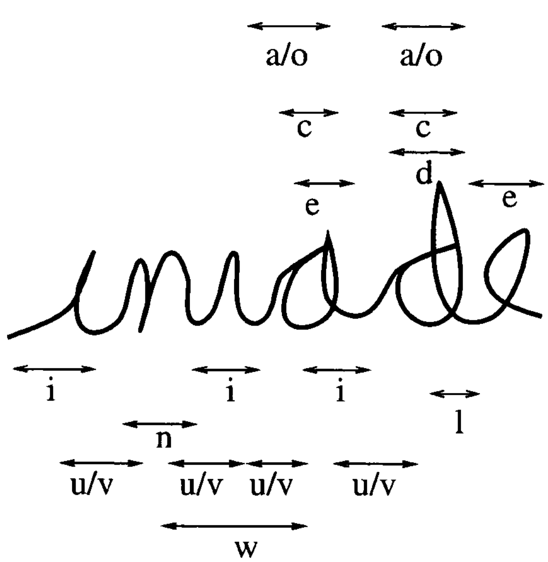


Fig. 14. Looking for instances of each character in a lexicon word (INVADE in this example) in the word image.

3.3 Holistic Approaches

Holistic approaches do not attempt to label parts of the image using sets of models; instead they extract holistic features from the word image and use the features directly to arrive at the word identity. In order for this feature-level matching to be possible, every candidate from the lexicon must have a feature representation similar to that used to represent the image features. The process of constructing a lexicon in which each lexicon entry is represented by its holistic features, or statistics about holistic features in the case of probabilistic methods, is sometimes referred to as "inverting the lexicon." Holistic methods described in the literature have used a variety of holistic features, representations, and matching methodologies.

Fig. 10 can be used to illustrate this point. Let us assume that our holistic features are {length, number of ascenders, number of descenders}. The lexicon can be inverted as follows:

```
MAIN [4 0 0]
GREENFIELD [10 3 1]
MASSACHUSETTS [13 3 0]
BUFFALO [7 3 2]
```

Note that the word "BUFFALO," written in uppercase, does not lend itself to holistic features. Assuming that we can derive the features from the shape of the words, the task of recognition, given the inverted lexicon, is quite trivial.

3.4 Remarks

Model-based approaches transform the problem of modeling the variability in the signal at the word-level, to modeling it at the level of submodels. The choice of submodels is critical, because of the assertion that these are identical irrespective of where they occur in words. The choice of characters as submodels may be the most obvious, but it is not the only one. It is difficult to capture all of the variability of handwritten words in terms of 26 submodels, especially when the shape of a character is a function of its neighbors (coarticulation effects).

Clearly, if the number of classes were finite and large amounts of training data were available, building a separate model for each word directly from the features would yield the best classification, since it would not be constrained by submodels. In practice, these conditions are met only when the lexicon is small and static.

So far, we have skirted the issue of how features differ from submodels. At the lower levels of the hierarchy are simple structures such as edges and other features of the pattern which are grouped into increasingly complex visual entities which approach characters in visual complexity at higher levels of the hierarchy. A variety of these visual entities, both ones that appear in the human visual system as well as arbitrary others, of increasing visual complexity may be imagined that span the spectrum from image pixels at one end to the word identity at the other.

We draw a line somewhere in the middle of this spectrum and call the visual entities preceding it "holistic features" and the ones following it "subword models." This distinction is necessarily a subjective one and is difficult to make unambiguously in some cases. Examples of word features are pixel density distributions, vertical and horizontal strokes, and perceptual features. Examples of subword models are the letters themselves, *graphemes* and other pseudoletter segments that result from an image-based segmentation scheme, and *n-grams* (letter combinations).

Although holistic and analytical approaches are commonly distinguished by the observation that the latter are "segmentation-based," the fact is that holistic and analytical paradigms comprise a continuum of approaches to word recognition. As noted by Casey and Lecolinet [7] in their survey of cursive word recognition, some form of segmentation is involved in all pattern recognition methods, which for holistic methods is their feature extraction phase.

> The main difference lies in the level of abstraction of the segmented elements: features (that is to say low-level elements) in the case of holistic methods, versus pseudo-letters in the case of analytical methods.

We have described the distinction between static and dynamic lexicon scenarios in Section 1.2, Figs. 3, and 4. Several examples of holistic recognizers that work with dynamic lexicons have been developed for use in the postal handwritten address interpretation context and described in the literature [11], [22], [23], [34]. Most systems used in the check recognition application [28] work with static lexicons.

4 Overview of Holistic Methods

The potential role of a holistic approach in a particular application scenario is inevitably linked to the size and static/dynamic nature of the lexicon. Examples of systems for classification, lexicon reduction, and verification are presented in this section.

Holistic methods can be categorized as follows:

1. *Domain:* online and offline.
2. *Lexicon use:* static and dynamic with applications in lexicon reduction and verification.
3. *Features:* low-level, intermediate-level, and high-level.
4. *Feature representation:* vectors, assertions, sequences, and graphs.
5. *Hybrid Methods:* Methods that explicitly use a combination of methodologies.

We will survey several methods that qualify as holistic methodologies by our definition (Section 3) and group them in the above categories. Clearly, methods will belong to several categories among the items listed above. We have attempted to highlight certain aspects of these methods under the various categories.

4.1 Domain

The earliest applications of the holistic paradigm were developed in the 1960s and 1970s for online HWR where the word was written on an electronic tablet, or on a screen with a lightpen. In this section, we use these efforts as starting points to cover the landscape of holistic approaches to offline HWR and we will refer to them throughout this section.

The whole word matching approach seems to have been first explored by Frishkopf and Harmon at Bell Labs in 1961 [15]. Words written on an electronic tablet are represented by the sequence of local x and y extrema along the trace of the word and compared with similarly encoded lexicon entries by looking for contiguous subsequences of similar extremes.

Around the same time, Earnest [12] at the Mitre Corporation designed a lexicon filter which used just the counts of ascenders and descenders and the presence or absence of a t-bar (i.e., global features) to identify similar words in a 10K lexicon.

Farag [14] represented the entire trace of the word by its chain coded representation. The approach involved extracting an 8-directional code sequence from the cursive input and using a first or second order Markov chain for recognition from a small, static lexicon of words.

Brown and Ganapathy [5] developed a system wherein a fixed 2D grid is imposed on the word image and the number of features (cusps, extensions, etc.) in each grid element counted. The resulting feature vector was compared with stored exemplars obtained from training using a Euclidean metric and k-NN is used to determine the final class.

The method presented by Miller [38] involved segmenting the online trace of a cursive word and classifying each of the stroke segments as one of a set of segments (codebook), obtained from unsupervised clustering of training words. An angular metric was used to rank a small static lexicon of three-character assembly language opcodes.

More recently, a highly accurate word recognition system that uses noncharacter specific features has been described [10]. The authors talk of the intent of the writer as one of conveying a complete message and not necessarily being careful about the individual characters. They describe the use of features such as cusps, crosses, dots, and breaks.

4.1.1 Remarks

Whatever may be the arguments put forth by developers of purely analytical word recognizers [13], holistic methods have been implemented and successfully used in practical systems [10].

4.2 Lexicon Use

The work of Bertille et al. [31] for recognition of unconstrained offline words in a small *dynamic* lexicon context is inspired by the earlier work of Salome et al. [32] in recognizing check amounts with a small static lexicon. Following segmentation of the word, loops, ascenders, and descenders are extracted and quantified into a small number of levels. Different symbols are assigned to each possible combination of loops and extensions that may be discovered within a segment, to yield a set of 27 symbols. A symbol descriptor for the word is thus obtained. Letters and common pseudoletters produced by segmentation (such as the first half of "n") are represented by a total of 65 three-state models. Given a training word and its corresponding descriptor, a word model for the word is constructed by concatenating letter and pseudoletter models and trained on a set of training strings (descriptors). The system achieves top choice accuracies of 91.2 percent, 77.6 percent, and 42.5 percent with dynamic lexicons of size 10, 100, and 1,000, respectively.

Holistic methods have been used for reduction of large lexicons, mainly in the online handwriting and printed domains. Perceptual features, especially, have been used internally by many analytical classifiers to rapidly discard dissimilar lexicon entries.

Verification of handwritten phrases may be thought of as the task of verifying that a given image of a word or phrase is that of a given ASCII string (or one of a given set of ASCII strings), frequently the result of another recognition algorithm.

The online system of Bramall and Higgins [1] is one of many that employ global features implicitly for lexicon reduction. The "candidate word hypothesization" phase of the system involves the use of features such as length, counts of features, and relative positions of ascenders and descenders to reduce a large static lexicon of 20,000 words lexicon to 184 on the average with 92 percent accuracy.

Global features of the word shape such as length and the presence of ascenders and descenders are useful for detecting unlikely matches in the lexicon, either explicitly as a lexicon filter, or implicitly as part of a word classifier. Earnest's lexicon filter for online script [12] described earlier, for example, uses just the counts of ascenders and descenders and the presence or absence of a t-bar to identify similar words in a 10K lexicon.

4.2.1 Remarks

For a holistic classifier, lexicons and training are tightly interwoven, since the lexicon entries are exactly the classes to be distinguished. Most of the literature deals only with small, fixed lexicons; in these cases, enough samples of each class are available to train the classifier in the conventional sense [5], [14]. This is clearly impractical in the case of large or dynamic lexicons.

In the latter case, it becomes necessary to obtain feature vectors for the lexicon entries via other means. In the machine print domain, it is possible to synthetically generate training samples of various fonts and sizes and even model forms of distortion [24]. Unfortunately, this method cannot be applied to unconstrained handwriting, owing primarily to the wider variety of handwriting styles and defects (breaks, fragmented strokes, skew, slant, open, and filled up loops) which are beyond the scope of existing models of handwriting. When the style of handwriting is constrained (to be online cursive, for example) and the features extracted are coarse, it may be possible to define production rules to determine whether an image descriptor derived from the image can be generated from a given lexicon entry [16]. For unconstrained handwriting, coarse holistic features such as ascenders, descenders, and length of a lexicon entry can be predicted from the features of the constituent characters using heuristic rules [33]. In these cases, training is in the form of heuristics or production rules being used to synthesize feature vectors corresponding to lexicon entries.

4.3 Features

There has been extensive research in the design of features for the recognition of isolated characters, which may be in theory applied to the recognition of entire words. Pixel-based features such as template correlation, transformations, and series expansions; features based on distribution of pixels derived from zoning, moments, n-tuples, characteristic loci, crossings, and distances; and low-level geometrical and topological features, such as strokes and curves in various directions, end points and intersections, and properties of the contour have been studied and extensively reviewed [47].

Although easy to extract and fairly insensitive to noise, features based on pixels or their distributions tend to be dependent upon position alignment and highly sensitive to distortion and style variations.

The last category of geometrical and topological features is by far the most popular for isolated handprinted characters, owing to their higher tolerance for distortion and stylistic variations and certain affine transformations. They form the lower tiers of a continuum of *structural* features (so named because they describe the characteristic geometry and topology of the word) that have been used for holistic recognition of words.

Fig. 12 can be referred to once again to illustrate the gradations of features from the fine (low-level) to the coarse (high-level).

4.3.1 Low-Level

Highly local, low-level structural features such as stroke direction distributions [45] have been applied successfully for holistic recognition of machine printed words. Hull et al. [54] experimented with both stroke direction distributions as well as *local shape templates* detected by convolution and thresholding [26]. In fact, they were found to perform better than either pixel-based features or higher level structural features such as perceptual features, whose detection is often unreliable [24]. Structural features at this level, however, are generally unsuitable for offline HWR, on account of wide variation in style.

Farag's method for online HWR [14] may also be thought of as using low-level features since the entire trace of the word was represented as an 8-directional chain code.

4.3.2 Intermediate-Level

Structural features at the intermediate-level include edges, end-points, concavities, diagonal and horizontal strokes, and exhibit a greater abstraction from the image (pixel or trace) level. The cusps and extensions extracted by the method of Brown and Ganapathy [5] and the local extrema extracted by Frishkopf and Harmon [15] may also be classified as intermediate-level structural features.

Dzuba et al. [10] describe a holistic word recognizer that works with a whole word or a phrase. They use features that reflect the importance of vertical extremas.

Guillevic and Suen [18] describe a feature-based holistic method for check recognition. For a training word of length n, a grid with n equal columns is used to capture ascender, descender and midzone loop positions (extracted from the contour) in the form of an n-bit vector. Strokes in the vertical, horizontal, and diagonal directions are extracted using morphological operators. The final feature vector for each training word is the concatenation of these binary vectors, along with counts of ascenders, descenders, loops, and length measured as the number of center-line crossings. Given a test word, the features are extracted using positions relative to the horizontal extent of the image. The authors report a top choice accuracy of 72 percent with a static check amount lexicon of 32 words.

Olivier et al. [42] take a structural description approach for holistic recognition of words from a static check amount lexicon. The center line intersects the thinned representation of the word at anchor points and divides it into structural primitives such as *upper loop* and *lower connection* (eight in all). The authors refer to these primitives as "strokes." Strokes sharing an anchor point taken together constitute a "grapheme." A set of 42 graphemes is obtained from all the graphemes found in a training set of words by an unsupervised clustering procedure. An image may now be represented as a sequence of either strokes or graphemes. The top choice of the stroke-based classifier, grapheme-based classifier, and their combination is reported to be 34 percent, 70 percent, and 72 percent, respectively.

4.3.3 High-Level

Perceptual features such as ascenders, descenders, loops, and length are easily perceived by the human eye, and we have reviewed evidence for their use in human reading. They are by far the most popular for holistic recognition of handwritten words.

Ascenders and descenders, while of uniform height and relatively easy to detect in machine print, are heavily subject to vagaries of style in handwriting, making their accurate detection a challenge. In theory, ascenders and descenders may be extracted by looking for parts of the word in the upper and lower zones, respectively. This, in turn, entails accurate reference line determination, which often fails in the presence of large skew, uneven writing, curved baseline, and for "top-heavy" images (e.g., "Falls"). Ascenders and descenders may also be detected directly from a run-length or contour representation.

Dots and holes may be computed by connected component analysis or alternatively by chain code analysis. Some features such as diagonal strokes and arcs may be easier to extract from the skeletonized image [24].

Word length is a particularly important perceptual feature [30] and may be estimated in the online case from the number of times the script traverses the "center line" as the ratio of this number to a statistic representing the number of traverses of the center line per letter of the average English word [4]. This method extends itself readily to offline script, but, in practice, the accuracy of the estimate is not satisfactory. Of course, the number of center-line crossings may be used in its raw form as a measure of length and compared with the estimated number of crossings for a given lexicon word. Other notions of length include the number of lower contour minima, the number of vertical strokes, and the number of possible segmentation points (ligatures and breaks).

Earnest's lexicon filter [12] which used counts of ascenders and descenders and the presence or absence of a t-bar is an early example of the use of perceptual features in a holistic HWR method. Simon and Baret [49] describe an approach to cursive script recognition that involves decomposing a cursive word into a pseudoperiodic signal (regular features) modulated by nonperiodic signals (irregular features). The irregularities are in essence perceptual features.

O'Hair and Kabrisky [41] describe the use of two-dimensional low frequency Fourier coefficients as features for holistic recognition of printed text. The low frequency components contain enough general information to uniquely identify the word from a fixed lexicon of possible words, but not the specific details of font and style. The latter are encapsulated by the high frequency components, which are ignored in the match. It is not clear that this approach will succeed with handwriting, given the large scale variations in writing style.

Miller's approach [38] of segmenting the online trace of a cursive word and classifying each of the stroke segments using a codebook is an example of an approach that uses a segmentation scheme in conjunction with a simple set of segment categories. These models are often derived from or are compositions of medium or high-level structural features. These methods may be classified as being holistic or analytical depending on the subjective decision as to whether the segments are complex enough to be called models.

4.3.4 Remarks

To summarize, the features best suited for holistic recognition of handwriting, as apparent from these studies, are higher level structural features, such as edges and end points, and perceptual features, such as dots, holes, ascenders, descenders, and t-bars. A particularly important perceptual feature is word length and many measures of length may be envisaged. The algorithmic accuracy of detection of perceptual features depends on the style and neatness of writing.

4.4 Feature Representation

The scheme used to represent holistic features is clearly a function of the features themselves and whether they are

low-level, medium-level, or high-level. Here, we review the predominant representation schemes.

4.4.1 Feature-Vectors and Matrices

Feature-vector representations are commonly used to represent low-level or intermediate-level features. The image is divided into sections using a fixed or variable grid, features are extracted from the sections, and the counts of different features in different sections are represented as a Boolean, integer or real-valued vector, or a matrix. Representing high-level features as a feature vector is less common.

The sectioning scheme is a form of implicit segmentation of the image or its representation and is often as important as the features themselves. The feature extraction of Brown and Ganapathy [5] divided the image into n equal sections and resulted in a 138-dimensional feature vector. Hull et al. [54] used a variable grid based on reference lines detected in the image.

Pixel-level features are uniquely identified by the x-y coordinates of the pixel. Low-level structural features are typically extracted by superimposing a rectangular grid on the image. This grid may either be fixed [5], or it may be variable (image-dependent) [24], [33].

For higher level structural features such as edges, end points, and perceptual features, the presence or absence of each feature is important and, consequently, the representation should allow matching of corresponding features in *nearby* cells as well. These features are generally represented more robustly by a graph or a string of codes, each code referring to a different feature or combination of features, although there are instances in the literature of binary feature vectors being used to represent the presence of higher level structural features in different sections of the word [18], [50].

4.4.2 Counts and Assertions

These are the simplest representation of high-level features. For example, Earnest's lexicon filter [12] used just the counts of ascenders and descenders and the presence or absence of a t-bar. Such simple features (sometimes called "global features") are often used to discard dissimilar word candidates from the lexicon.

4.4.3 Sequences

The word is represented as a sequence of symbols representing a set of structural primitives, which correspond to intermediate or high-level features or combinations of such features. This constitutes an implicit segmentation of the image into the structural primitives. Some hybrid methods explicitly segment the image and extract holistic features from the segments.

A *location coded string representation* tags each code with the "positions" in which it occurs, again with reference to a fixed or variable global reference frame. For instance, "O:256" may indicate that there are three holes in the image, located at the second, fifth, and sixth positions along the length of the image [24].

Since a word is approximately a one-dimensional signal that flows from left to right, a sequential representation of such codes may suffice as a description of shape. Accordingly, a *symbol string representation* denotes the image as a sequence of codes. Adjacency and relative locations of structural features of different types are readily captured by these descriptors [16]. Features that may be located above or below others are better described in separate strings.

Moreau [40] extracted vertical and horizontal strokes, loops, i-dots, and t-bars from the offline image to obtain a string descriptor and compared the descriptor with unique prototypes of words found in French check amounts and their more common orthographic deviations. The unique prototype for each class was obtained as the mode of the descriptors obtained from training samples of that class.

Salome et al. [32] extracted ascenders, descenders, loops, i-dots, and unattached t-bars from the contours of connected components and obtained a string descriptor. Word length was estimated as the number of letter segments obtained as a by-product of a separate analytical subsystem. The Levenshtein metric was used to compare the test string with reference strings obtained from training corresponding to a small lexicon of check amounts.

4.4.4 Graph Structures

The whole image may be represented by a graph with features as nodes and spatial relationships between them as the edges [17]. Graph representations are powerful in that they can represent both positions of features as well as relationships between them. Fig. 15 is an illustration of such a graph structure.

Paquet and Lecourtier [43] describe a check amount recognition method where the intersections of the middle line with the word ("guiding points") are first determined. Stroke following is initiated at each guiding point and each point is coded by features of the primitive stroke segments starting and ending at the point. The "graph" (essentially the thinned image) obtained from stroke tracing is analyzed into seven types of primitives (upper strokes, lower strokes, upper connection, etc.) and a symbol string describing the structure of the graph is achieved. The test string is compared with empirically obtained reference strings using the Levenshtein metric.

Camillerapp et al. [6] labeled singular vertices (end-points, crossings, and points of local curvature) in the skeletonized gray-level image and obtained a tree of stroke primitives. Each tree node was described by the type of primitive, vertical word zone position, and its relative horizontal position within the word. Each lexicon word was coded as a similar tree of primitives, except that each node could describe a set of primitives covering variations that may be expected at that point.

4.4.5 Remarks

The choice of a representation scheme depends on the implementation constraints and on the eventual matching strategy used. The types of features seem to be common in that they are not specific to individual characters. Statistical classification techniques use feature vectors, heuristic matching techniques predominantly use counts and assertions, symbolic matching methods primarily use lexicon coded strings, and graph representation methods naturally favor graph matching algorithms.

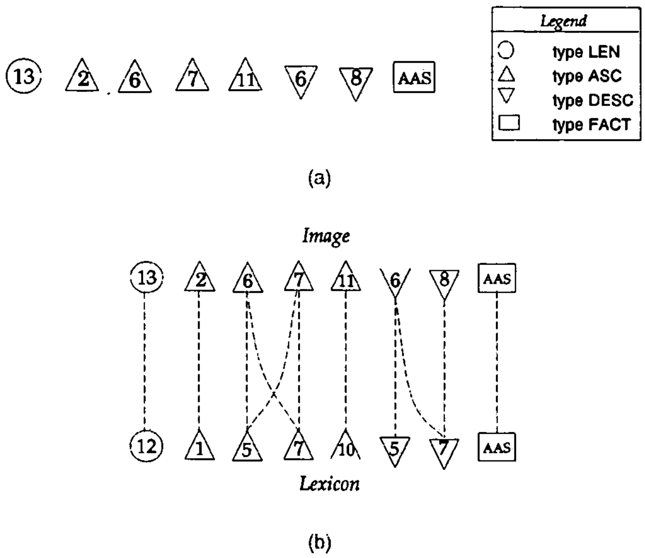


Fig. 15. (a) Wordgraph for the image of "Buffalo." (b) Possible node associations between the image and the lexicon entry `Buffalo`. A possible node association is denoted by a dotted arc between a node of the first graph and one of the second.

4.5 Hybrid Methods

Some methods adopt both analytical and holistic features. The commercial French check processing system of Simon et al. [50] uses seven operators to estimate the probability of each of the 25 lexicon classes. The first of these is based on comparing a structural description known as the "holography" of the test word with prototypes of each class and is based on Gorsky's earlier work [17]. The other six operators are based on different perceptual features:

1. length (number of segments obtained from the analytical method),
2. positions of ascenders and descenders,
3. positions of sets of overlapping horizontal segments,
4. mid-zone loops,
5. dots above the half-line, and
6. mid-zone crosses.

Dodel and Shinghal [9] describe a hybrid analytical-holistic method for offline words to identify the correct class from a static lexicon of 31 words. Aspect ratio (horizontal extent/midzone width) and relative positions of ascenders and descenders are used to achieve direct recognition of some words such as "eight" and partial recognition of others.

Hull et al. [54] estimate length of printed words from character segmentation and word case from reference lines and use these global features to filter the lexicon. Their "segmentation-based" approach is actually holistic since it involves segmenting the word image into characters and concatenating the pixels corresponding to each segmented character (normalized to a 24 x 24 grid) into a 24 x 24 x N vector, where N is the length of the word. They use the Baird templates [26] and stroke direction distribution features.

4.5.1 Remarks

It is natural for word recognition engines to consider a hybrid of recognizers for best performance. Analytical and holistic methods can complement each other's strengths and provide for a robust system.

5 Summary

The Holistic paradigm in handwritten word recognition is one that treats the word as a single, indivisible entity and attempts to recognize it using features of the word as whole, and is inspired by psychological studies of human reading, which indicate that humans use features of word shape such as *length, ascenders, and descenders* (see Fig. 5) in reading.

Holistic approaches circumvent the issues of segmentation ambiguity and character shape variability that are primary concerns for analytical approaches and may succeed on poorly written words where analytical methods fail to identify character content. However, their treatment of lexicon words as distinct pattern classes has traditionally limited their application to recognition scenarios involving small, static lexicons.

Given the difficulty of the task of reading handwriting, practical recognition engines must use multiple classification algorithms and complex strategies for combining classifier decisions and thresholding based on classification confidences for rejection of classification errors. In this survey, we have attempted to take a fresh look at the potential role of the Holistic paradigm in handwritten word recognition, in the light of this observation.

The Holistic paradigm draws inspiration from studies of individuals with acquired dyslexia, studies of reading development, and studies involving proofreading tasks which provide evidence for the existence of a parallel holistic reading process in both developing and skilled

readers; however, there appears to be no consensus on how word shape information is combined with letter identities.

We have attempted to characterize approaches to recognition as a continuous spectrum based on the visual complexity of the unit of recognition employed. Holistic features may be distinguished from subword models in the visual processing hierarchy by their relatively lower visual complexity; however, this distinction is subjective. A holistic approach may be defined as one which does not search for subword models. Analytical approaches are more accurately called model-based approaches.

Holistic systems generally adopt either a *feature-extraction* or a *structural description* approach to the problem of representing word shape. The features themselves may be classified broadly as being pixel-based or structural. Higher level structural features appear to be best-suited for holistic recognition of handwriting and are represented as feature vectors, location-coded and symbol strings, and graphs, to name a few common ones. The matching methodology adopted is related closely to the representation of features.

Holistic recognition systems are characterized by an integration of training and the lexicon, whose presence is often an implicit assumption in the design of holistic word recognition algorithms. Most implementations of holistic approaches in the offline HWR domain have been used for the classification of small, static lexicons. Lexicon reduction and verification of recognition results have recently emerged as other applications of the holistic paradigm.

Given the evidence from reading studies, the intrinsic advantage of computational economy, and orthogonality with respect to analytical approaches, we believe that the holistic paradigm holds immense promise for realizing near-human performance.

REFERENCES

[1] P.E. Bramall and C.A. Higgins, "A Blackboard Approach to On-Line Cursive Handwriting Recognition for Pen-Based Computing," *Proc. Third Int'l Workshop Frontiers in Handwriting Recognition (IWFHR-III)*, pp. 295-304, 1993.

[2] D.E. Broadbent and M.H.P. Broadbent, "General Shape and Local Detail in Word Perception," *Attention and Performance VI*, S. Dornic, ed., Erlbaum, 1977.

[3] D.E. Broadbent and M.H.P. Broadbent, "Priming and the Passive/Active Model of Word Recognition," *Attention and Performance VIII*, R.S. Nickerson, ed., Lawrence Erlbaum, 1980.

[4] E.R. Brocklehurst and P.D. Kenward, "Preprocessing for Cursive Script Recognition," *NPL Report*, Nov. 1988.

[5] M.K. Brown and S. Ganapathy, "Cursive Script Recognition," *Proc. Int'l Conf. Cybernetics and Soc.*, pp. 47-51, Oct. 1980.

[6] J. Camillerapp, G. Lorette, G. Menier, H. Oulhadj, and J.C. Pettier, "Off-Line and On-Line Methods for Cursive Handwriting Recognition," *From Pixels to Features III: Frontiers in Handwriting Recognition*, S. Impedovo and J.C. Simon, eds., pp. 273-288, North-Holland, 1992.

[7] R.G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 690-706, July 1996.

[8] J.M. Cattell, "The Time Taken Up by Cerebral Operations," *Mind*, vol. 11, pp. 220-242, 1886.

[9] J.-P. Dodel and R. Shinghal, "Symbolic/Neural Recognition of Cursive Amounts on Bank Cheques," *Proc. Third Int'l Conf. Document Analysis and Recognition (ICDAR)*, pp. 15-18, Aug. 1995.

[10] G. Dzuba, A. Filatov, D. Gershuny, and I. Kil, "Handwritten Word Recognition—The Approach Proved by Practice," *Proc. Fifth Int'l Workshop Frontiers in Handwriting Recognition*, pp. 99-112, 1998.

[11] A.C. Downton, E. Kabir, and R. Birch, "Recognition and Verification of Postcodes in Handwritten and Hand-Printed Addresses, *Proc. 10th Int'l Conf. Pattern Recognition*, pp. 469-473, 1990.

[12] L.D. Earnest, "Machine Recognition of Cursive Writing," *Proc. IFIP Congress*, pp. 462-466, 1962.

[13] S. Edelman, T. Flash, and S. Ullman, "Reading Cursive Handwriting by Alignment of Letter Prototypes," *Int'l J. Computer Vision*, vol. 5, no. 3, pp. 303-331, 1990.

[14] R. Farag, "Word Level Recognition of Cursive Script," *IEEE Trans. Computers*, vol. 28, no. 2, Feb. 1979.

[15] L.S. Frishkopf and L.D. Harmon, "Machine Reading of Cursive Script," *Proc. Fourth London Symp. Information Theory*, 1961.

[16] N. Nasrabadi, G. Seni, and R. Srihari, "An Online Cursive Word Recognition System," *Proc. IEEE Computer Vision and Pattern Recognition*, June 1994.

[17] N.D. Gorsky, "Off-Line Recognition of Bad Quality Handwritten Words Using Prototypes," *Fundamentals of Handwriting Recognition*, S. Impedovo, ed., pp. 199-217, Springer-Verlag, 1993.

[18] D. Guillevic and C.Y. Suen, "Cursive Script Recognition Applied to the Processing of Bank Cheques," *Proc. Third Int'l Conf. Document Analysis and Recognition (ICDAR)*, pp. 11-14, Aug. 1995

[19] R.N. Haber and R.M. Schindler, "Errors in Proofreading: Evidence for Syntactic Control of Letter Processing?" *J. Experimental Psychology: Human Perception and Performance*, vol. 7, pp. 573-579, 1981.

[20] L. Harmon, "Automatic Recognition of Print and Script," *Proc. IEEE*, vol. 60, no. 10, pp. 1165-1176, 1972.

[21] L. Henderson, *Orthography and Word Recognition in Reading*. Academic Press, 1982.

[22] A. Hendrawan and A.C. Downton, "Verification of Handwritten British Postcodes Using Address Features," *Fundamentals of Handwriting Recognition*, S. Impedovo, ed., pp. 313-317, Springer-Verlag, 1993.

[23] A. Hendrawan, A.C. Downton, and C.G. Leedham, "A Fuzzy Approach to Handwritten Address Verification," *Proc. Third Int'l Workshop Frontiers in Handwriting Recognition (IWFHR-III)*, pp. 207-216, 1993.

[24] T.K. Ho, "A Theory of Multiple Classifier Systems and Its Application to Visual Word Recognition," PhD thesis, State Univ. of New York at Buffalo, 1992.

[25] D. Howard, "Reading Without Letters?" *The Cognitive Neuropsychology of Language*, M. Coltheart, G. Sartori, and R. Job, eds., Lawrence Erlbaum, 1987.

[26] L.D. Jackel, H.S. Baird, H.P. Graf, and W.E. Hubbard, *A VLSI Architecture for Binary Image Classification*, pp. 275-186, North Holland, 1989.

[27] G.W. Humphreys, L.J. Evett, and P.T. Quinlan, "Orthographic Processing in Visual Word Recognition," *Cognitive Psychology*, vol. 22, pp. 517-560, 1990.

[28] G. Kim and V. Govindaraju, "Bank Check Recognition Using Cross Validation between Legal and Courtesy Amounts," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 11, no. 4, pp. 657-674, 1997.

[29] N. Lindgren, "Cursive Script Recognition," *IEEE Spectrum*, May 1965.

[30] R. N. Haber, L.R. Haber, and K.R. Furlin, "Word Length and Word Shape as Sources of Information in Reading," *Reading Research Quarterly*, vol. 18, pp. 165-189, 1983.

[31] J. Bertille, M. Gilloux, and M. Leroux, "Recognition of Handwritten Words in a Limited Dynamic Vocabulary," *Proc. Third Int'l Workshop Frontiers in Handwriting Recognition*, pp. 417-422, 1993.

[32] J.C. Salome, M. Leroux, and J. Badard, "Recognition of Cursive Script Words in a Small Lexicon," *Proc. First Int'l Conf. Document Analysis and Recognition (ICDAR '91)*, pp. 774-782, Sept./Oct. 1991.

[33] S. Madhvanath and V. Govindaraju, "Holistic Lexicon Reduction," *Proc. Third Int'l Workshop Frontiers in Handwriting Recognition (IWFHR III)*, pp. 71-81, 1993.

[34] S. Madhvanath and V. Govindaraju, "Holistic Lexicon Reduction for Handwritten Word Recognition," *Proc. Eighth SPIE Int'l Symp. Electronic Imaging: Science and Technology*, Jan./Feb. 1996.

[35] S. Madhvanath, G. Kim, and V. Govindaraju, "Chaincode Processing for Handwritten Word Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, Sept. 1999.

[36] S. Madhvanath, E. Kleinberg, and V. Govindaraju, "Empirical Design of a Multiclassifier Thresholding/Control Strategy for Recognition of Handwritten Street Names," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 11, no. 6, pp. 933-946, 1997.

[37] J.L. McClelland, "Preliminary Letter Identification in the Presentation of Words and Nonwords," *J. Experimental Psychology: Human Perception and Performance*, vol. 2, pp. 80-91, 1976.

[38] G.M. Miller, "Real-Time Classification of Handwritten Script Words," *Proc. IFIP Congress*, pp. 218-223, Aug. 1971.

[39] A.F. Monk and C. Hulme, "Errors in Proofreading: Evidence for The Use of Word Shape in Word Recognition," *Memory and Cognition*, vol. 11, pp. 16-23, 1983.

[40] J. Moreau, "A New System for Automatic Reading of Postal Checks," *Int'l Workshop Frontiers in Handwriting Recognition (IWFHR-2)*, pp. 121-132, Sept. 1991.

[41] M.A. O'Hair and M. Kabrisky, "Recognizing Whole Words as Single Symbols," *Proc. First Int'l Conf. Document Analysis and Recognition (ICDAR '91)*, pp. 350-358, Sept./Oct. 1991.

[42] C. Olivier, T. Paquet, M. Avila, and Y. Lecourtier, "Recognition of Handwritten Words Using Stochastic Models," *Proc. Third Int'l Conf. Document Analysis and Recognition (ICDAR)*, pp. 19-23, Aug. 1995.

[43] T. Paquet and Y. Lecourtier, "Handwriting Recognition: Application to Bank Cheques," *Proc. First Int'l Conf. Document Analysis and Recognition (ICDAR '91)*, pp. 749-757, Sept./Oct. 1991.

[44] G.M. Reicher, "Perceptual Recognition as a Function of Meaningfulness of Stimulus Materials," *J. Experimental Psychology*, vol. 81, pp. 274-280, 1969.

[45] K. Yamamoto, S. Mori, and M. Yasuda, "Research on Machine Recognition of Handprinted Characters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 4, pp. 386-405, July 1984.

[46] K.M. Sayre, "Machine Recognition of Handwritten Words: A Project Report," *Pattern Recognition*, vol. 5, pp. 213-228, 1973.

[47] L. Schomaker and E. Segers, *Advances in Handwriting Recognition*, S.-W. Lee, ed., vol. 34, World Scientific, 1999.

[48] P.H.K. Seymour, "Developmental Dyslexia," *Cognitive Psychology: An International Review*, M.W. Eysenck, ed., John Wiley and Sons, 1990.

[49] J. Simon and O. Baret, "Formes Regulieres et Singulares; Application a la Reconnaisance de L'ecriture Manuscrite," *C.R. Acad. Scr. Paris*, vol. 309, pp. 1901-1906, 1989.

[50] J.C. Simon, O. Baret, and N.D. Gorski, "A System for the Recognition of Handwritten Literal Amounts of Checks," *Proc. IAPR Workshop Document Analysis Systems*, A.L. Spitz and A. Dengel, eds., pp. 265-287, 1994.

[51] S.J. Soltysiak, "Visual Information in Word Recognition: Word Shape or Letter Identities?" *Proc. Workshop Integration of Natural Language and Vision Processing*, 1994.

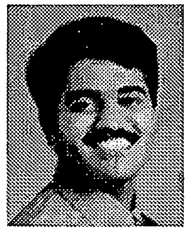
[52] C.C. Tappert, C. Suen, and T. Wakahara, "The State of the Art in On-Line Handwriting Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, Aug. 1990.

[53] J. Theios and J.G. Muise, *The Word Identification Process in Reading*, vol. 2. Hillsdale, N.J.: John Wiley & Sons, 1977.

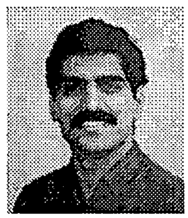
[54] J.J. Hull, T.K. Ho, and S.N. Srihari, "Word Recognition with Multilevel Contextual Knowledge," *Proc. First Int'l Conf. Document Analysis and Recognition (ICDAR '91)*, pp. 905-915, Sept./Oct. 1991.

[55] G. Underwood and K. Bargh, "Word Shape, Orthographic Regularity, and Contextual Interactions in a Reading Task," *Cognition*, vol. 12, pp. 197-209, 1982.

[56] D.D. Wheeler, "Word Recognition Processes," *Cognitive Psychology*, vol. 1, pp. 59-85, 1970.



Sriganesh Madhvanath received the BTech degree in computer science from the Indian Institute of Technology, Bombay, in 1990 and the MS and PhD degrees from the State University of New York at Buffalo, in 1993 and 1997, respectively. He worked as a research assistant at the Center for Excellence in Document Analysis and Recognition (CEDAR) from 1991 to 1996. He then joined the Image and Multimedia Systems Group at the IBM Almaden Research Center, San Jose, California as a member of the research staff. His research interests include classifier decision combination, genetic algorithms, and document analysis and recognition. He is a member of the IEEE.



Venu Govindaraju received his PhD in computer science from the State University of New York at Buffalo in 1992. He has coauthored more than 100 technical papers in various international journals and conferences and has one US patent. He is currently the associate director of CEDAR as well as an associate professor in the Department of Computer Science and Engineering at the University of New York. He is the associate editor of the *Journal of Pattern Recognition* and the area chair of the IEEE SMC technical committee for pattern recognition. Dr. Govindaraju has been a coprincipal investigator on several federally sponsored and industry sponsored projects. He is presently leading multiple projects on postal applications. He is the program cochair of the upcoming International Workshop on Frontiers in Handwriting Recoignition in 2002. He is a senior member of the IEEE.

A multi-classifier combination strategy for the recog of handwritten cursive words

Plessis, B. Sicsu, A. Heutte, L. Menu, E. Lecolinet, E. Debon, O. Moreau, J.-V.
MATRA CAP Systemes, Saint-Quentin-en-Yvelines, France;

This paper appears in: **Document Analysis and Recognition, 1993., Proce the Second International Conference on**

Meeting Date: 10/20/1993 - 10/22/1993
Publication Date: 20-22 Oct. 1993
Location: Tsukuba Science City Japan
On page(s): 642 - 645
Reference Cited: 8
Inspec Accession Number: 4951011

Abstract:
A **recognition** scheme for reading handwritten cursive words using three wor **recognition** techniques is described. The focus is on the implementation used combine the three techniques based on a comparative study of different strate first **holistic recognition** technique derives a global encoding of the word. Th techniques both rely on the segmentation of the word into letters, but differ in character classifier they use. The former runs a statistical linear classifier, and runs a neural network with a different representation of the input data. The te comparison, and combination studies have been performed on word images fr provided by the USPS. The top choice **recognition** rates achieved so far corre 88%, 76%, 65% with respect to lexicon sizes of 10, 100, and 1000 words

Index Terms:
character classifier **handwriting recognition** handwritten cursive words **holistic rec** technique image segmentation lexicon sizes mail multiclassifier combination strateg nets neural network optical character **recognition** reading **recognition rates** **reco** scheme statistical linear classifier word **recognition** word segmentation character c **handwriting recognition** handwritten cursive words **holistic recognition** technique segmentation lexicon sizes mail multiclassifier combination strategy neural nets ne network optical character **recognition** reading **recognition** rates **recognition** sche statistical linear classifier word **recognition** word segmentation

Documents that cite this document

There are no citing documents available in IEEE Xplore at this time.

Search Results [PDF FULL-TEXT 336 KB] DOWNLOAD CITATION

Zoning invariant wholistic recognizer for hybrid recognition of handwriting

R.K. Powalka, N. Sherkat, R.J. Whitrow

The Nottingham Trent University, Department of Computing
Burton Street, Nottingham NG1 4BU, United Kingdom
e-mail: {rkp, ns, rjw}@doc.ntu.ac.uk

Abstract

This paper describes a wholistic recognizer developed for use in a hybrid recognition system. The recognizer uses information about the word shape. As this information is strongly related to word zoning, care is taken to avoid limitations resulting from the inaccuracy of zone detection. The recognizer uses a very simple set of features and a fuzzy set based pattern matching technique. This aims to increase its robustness, but also causes problems with disambiguation of the results. A verification mechanism, using letter alternatives as compound features, is introduced. The letter alternatives are obtained from a segmentation based recognizer coexisting in the hybrid system. The wholistic recognizer is found capable of outperforming the segmentation based one, despite the remaining disambiguation problems. When working together in a hybrid system, the results are significantly higher than those of the individual recognizers. Recognition results are reported and compared.

Keywords: on-line cursive script recognition, word shape, wholistic recognition, multiple interactive segmentation, hybrid recognition, results combination

1 Introduction.

There appears to be no single, uniform approach to the problem of handwriting recognition. Various developed algorithms achieve considerable success with certain handwriting styles, but cannot maintain their high recognition rates for other styles. On-line cursive script recognition system developed by the authors [5][6] is not an exception. It is capable of recognition rates exceeding 90%, as well as rates lower than 30%, depending on the writer.

It is found that various recognition methods produce different results and errors. Their decisions are frequently complementary. Combining results of multiple recognizers provides nearly universal improvement (e.g. [4], [6], [8]).

A recognition system developed by the authors uses multiple interactive segmentation [5], a segment and recognize approach. The method works well in cases where correct segmentation is possible, however it inherently fails for illegible writing. Word ending postulation [6] has been introduced to cope with words becoming illegible towards their endings. Wholistic recognition approach has also been adopted as a more general solution. Where correct segmentation is possible, the multiple interactive segmentation recognizer provides an answer. This answer can be further verified by a wholistic recognizer. Where the segmentation fails, the wholistic recognizer is relied upon. An ascender/descender word shape recognizer has been introduced for such a purpose [7]. The performance of the ascender/descender recognizer is significantly limited by the accuracy of identification of the ascenders and descenders within the word which depends on the correct estimation of the word zoning. The accuracy of the zoning information extraction is still far from desired [3][7]. However, a hybrid system provides an universal improvement [7].

The present paper discusses a new on-line wholistic recognizer designed for use in a hybrid recognition system, beside the multiple interactive segmentation recognizer.

2 Vertical bars recognizer

The vertical bars recognizer uses parts of the pen trajectory where it was moving downwards and approximately vertically. Such parts are referred to as *vertical bars*. Their number, height and position are the features used for the word shape matching. Zoning information is not required. The pattern of vertical bars representing an unknown word is compared to all the appropriate patterns within the database. As a result the vertical bars recognizer provides better recognition rates, especially when dealing with data difficult to zone reliably. By matching all the located bars, including those in the middle zone, the recognizer implicitly makes use of the information present in the middle zone.

2.1 Word shape encoding

Word shape encoding using vertical bars is presented in Figure 1. Preprocessed ink data points are analysed and local vertical extrema are located (Figure 1a). Vertical bars are located using the local extrema. Each pair maximum/minimum within a single stroke results in a vertical bar (Figure 1b). The order of the extrema is significant as the encoding is designed to represent only downwards directed parts of the pen trajectory.

The encoding process results in a number of bars of different height and position (Figure 1b), usually different from the ideal encoding (Figure 1c). The better a word is written, the more its bar encoding resembles the ideal.

0-8186-7128-9/95 $4.00

2.2 Word shape matching principles

Word shape matching is performed by calculating the distance between the pattern derived from the data and all the relevant database patterns. Relevant database patterns are identified according to the number of vertical bars. In order to make a comparison manageable, the vertical bar pattern derived from the data must be normalised. Area of the highest ink density within the word is used.

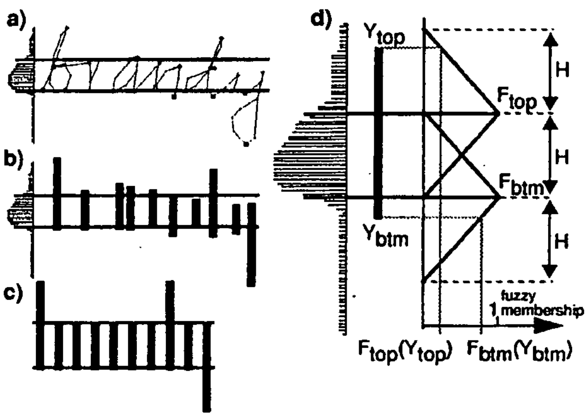


Figure 1: Word shape matching principles: a) original word; b) vertical bar encoded word; c) database pattern for the word; d) fuzzy set based encoding of vertical position of bar endpoints.

Figure 1 presents principles of word shape matching using vertical bar coding. The comparison process takes into account bar sizes and vertical positions. These two parameters are equivalent to the vertical position of top and bottom of each bar (Y_{top} and Y_{btm} in Figure 1d). The horizontal bar position is not used.

The position of endpoints of each vertical bar are encoded using two fuzzy sets [9], as explained in Figure 1d. The obtained fuzzy membership values can be directly compared to values stored in the database. The distance between compared patterns is defined as the sum of all the distances between appropriate individual bars:

$$Dist = \sum_{i=1}^{n} D_i \qquad \text{(EQ 1)}$$

where $Dist$ is the distance between patterns, D_i is the distance between i-th bars and n is the number of bars within the pattern.

Distance between individual bars is calculated as follows:

$$D = max(D_{top}, D_{btm}) \qquad \text{(EQ 2)}$$

where max function implements AND of differences of fuzzy conditions, D is the distance between individual bars, D_{top} and D_{btm} are differences between vertical positions of the compared patterns (top and bottom of bars, respectively):

$$D_* = |F_*(Y_*) - Fdb_*| \qquad \text{(EQ 3)}$$

$F_*(Y_*)$ is the fuzzy position of bar endpoints in the data pattern (Figure 1d), and Fdb_* is the fuzzy position of bar endpoints in the database pattern (top and bottom, respectively), as depicted in Figure 3.

2.3 Word shape matching algorithm

The distance calculated in the previous section (EQ 1) requires the number of vertical bars in the compared patterns to be identical. This is the simplest, ideal case. In reality however, variations in handwriting style may result in superfluous vertical bars (see Figure 2). In such cases the bar encoding derived from the data contains too many bars, which need to be dealt with. An attempt may be made to remove the superfluous bars. These can however be often confused with significant bars, resulting in removing important information. Another possibility is to make the matching algorithm tolerant towards extra bars within the pattern encoding the input data.

Superfluous bars have been observed to directly precede ligatures in majority of cases within the collected handwriting data. This allows to reduce the complexity of the comparison. Database patterns contain information about the position of ligatures (see Section 3). During the comparison process, ligature positions in the database pattern may be assigned bars from the data pattern. This way the data pattern bars are ignored. For each ligature position two possibilities are investigated: the ligature is simply skipped, or it absorbs one (presumably superfluous) bar of the data pattern. The match resulting in the lowest distance between patterns is retained. Figure 2 presents examples of the vertical bar matching process where data samples contain superfluous bars (indicated in the figure).

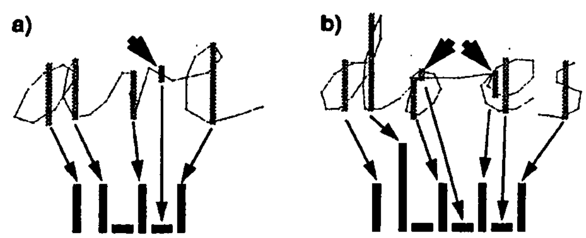


Figure 2: Vertical bar pattern matching: a) regular situation; b) irregular situation: two consecutive superfluous bars. Arrows represent the best match, horizontal bars represent ligatures.

Comparisons where no bars are ignored are preferred, as all the information within patterns is used. Hence each ignored bar increases the distance between patterns. The distance function EQ 1 is extended in order to accommodate that:

$$Dist = \sum_{i=1}^{n} D_i + \sum_{j=1}^{m} Pd_j \qquad \text{(EQ 4)}$$

where Pd_j is the distance penalty incurred by the j-th ignored bar and m is the total number of ignored bars within the data pattern. The distance penalty for vertical bars in the data pattern is calculated as follows:

$$Pd = \frac{h}{H} \qquad \text{(EQ 5)}$$

where h is the height of the bar and H is the height of the zone of the highest ink density within the word (see

Figure 1d). The function is designed to be proportional to the size of the discarded bar.

3 Generation of word shape database

A database of word shape information is necessary for the word shape recognition. It is created using a priori information about letters and words.

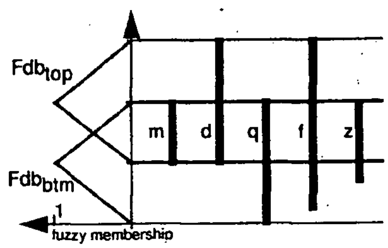


Figure 3: Vertical bars used by word shape database.

Each letter is represented as a sequence of vertical bars. Five types of vertical bars are used (Figure 3). Three of them (m, d and q) are obvious. Two remaining (f and z) are introduced to cope with different ways of writing letters "f" and "z." Bar f nearly reaches the lower zone line. It will provide better match for letters "f" which span all three zones. Bar z ends near the middle zone. It will provide better match for letters "z" which appear in the middle zone only. The mechanism for dealing with superfluous bars in data patterns allows extra bars only at the end of each letter. For this reason a special "ligature" symbol is inserted between bar patterns for each letter.

For a purpose of word length comparison and letter verification a letter width metric is also used. Letter widths are expressed in terms of the middle zone height. The following widths are used: 0.5 ("fijlrt"), 0.7 ("v"), 1.0 ("abcdeghknopqsuxyz"), 1.3 ("w") and 1.5 ("m").

Database patterns generated using the described model are not expected to ideally match vertical bar patterns obtained from the real data.

4 Letter verification

Due to a choice of a very simple set of features and a relaxed matching algorithm used, the vertical bars recognizer is frequently capable of correct recognition, but has difficulties choosing the correct alternative as the best answer amongst other words of similar shape. Additional features like loops, arcs, cusps, concavities and convexities (e.g. a set of features described in [2]) could be used to make the recognizer more discriminative. Another possibility is to use letter alternatives located and recognized by the multiple interactive segmentation recognizer [5]. Letters are combinations of the mentioned features, hence they could be used as compound features.

4.1 Matching word with letter graph

A directed acyclic graph of letter alternatives is obtained from the multiple interactive segmentation recognizer. The graph is recursively searched for consecutive letters within the word alternative being verified. Any number of letter alternatives are allowed to be missing. Scores of the located letter alternatives are averaged to provide the score of the letter verification process. In effect, the more letters are located, and the higher their scores are, the better is the letter verification score.

Scores obtained both by the vertical bars recognizer and the letter verification are averaged into a single score.

4.2 Lowering computational intensity of the letter verification

Due to the nature of the multiple interactive segmentation process and the letter recognizer used by the multiple interactive segmentation [6], the generated graphs of letter alternatives can often be large. As any number of letters can be allowed to be missing, analysing large letter graphs can be computationally intensive.

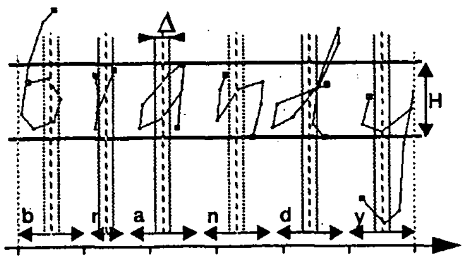


Figure 4: Limiting positions of letters in the letter verification process. The verified word alternative is "brandy."

In order to lower the computational intensity of the letter verification, the position at which each letter alternative is allowed within the word is limited. A set of letter position limits is derived for each word alternative considered in the recognition process. Figure 4 presents the set of letter position limits derived for the word alternative "brandy" which is the correct recognition result. The expected letter position is calculated using letter width metrics (see Section 3). Zones of the width Δ are placed in the middle of the expected position of each letter. Letter alternatives considered in the letter verification process have to intersect with the relevant zone. Otherwise they are discarded.

Tests show that the imposed limitation significantly reduces the processing requirements of the letter verification. The saving depends on the word length. A 42-fold saving was observed for the word "brandy" in Figure 4. Savings are still larger for longer words.

5 Results

Tests were performed to assess the performance of the developed recognizer. Handwriting data were collected from eighteen writers. Each writer wrote 200 words, one to sixteen letters long. An NCR 3125 pen computer was used. Sixteen writers are right-handed, two are left-handed. Thirteen writers have had no previous experience with the "electronic paper." A medium size lexicon of 4107 words was used. No specific criteria for selecting words of the lexicon were used.

Figure 5 presents results obtained for various recognizers and their varieties:

- multiple interactive segmentation (SEG). Results are provided for comparison only;

- vertical bars (VB);
- vertical bars with letter verification (VBLV);
- hybrid SEG x VB (HYB1);
- hybrid SEG x VBLV (HYB2).

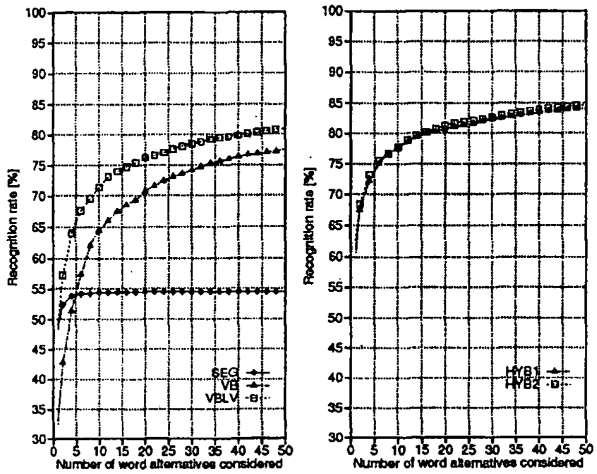


Figure 5: Recognition rates obtained for various recognizers and their combinations.

Recognition results are averaged over all writers. The results are provided for various number of word alternatives taken into account. When only the best word alternative is taken into consideration, SEG performs best amongst the individual recognizers. VBLV performs marginally worse. VB performs significantly worse. As the number of results alternatives taken into consideration grows, the wholistic recognizers outperform SEG.

Hybrid recognizers provide a universal improvement. An improvement of 13% is observed for the top word alternative. As the number of word alternatives taken into account increases, the recognition rate of the hybrid recognizers is 5 to 10% better than that of VBLV. When compared to SEG, a 20 to 30% improvement in the recognition rate is observed. Interestingly, HYB1 and HYB2 provide nearly identical results.

Significant increase in the recognition rate for a larger number of alternatives indicates that the results of the wholistic and hybrid recognizers can be further improved.

6 Discussion

The vertical bars recognizer described in this paper is an improvement over the previously reported wholistic recognizer [7]. Fuzzy sets are used to represent the position of the vertical bars ends. Hence no zoning classification of vertical bars is necessary. The use of fuzzy sets also allows special treatment of letters with different forms, like "f" or "z." Smaller vertical bars, present in the middle zone, are also used in the matching process, which is an advantage over the previously reported wholistic recognizer. Larger vertical bars have been observed to be relatively invariant within words. Smaller bars, which occur in the middle zone, tend to have higher variability (see Section 3). The matching algorithm introduced to cope with this problem is comparable to dynamic programming.

A conscious decision was made to strictly limit the number of features used. The resulting system has bigger difficulty choosing the correct word alternative as the best answer, however it is expected to reject the correct alternative less frequently.

The resulting disambiguation problem is addressed by letter verification. The treatment of letter alternatives as higher level handwriting features is also reported in the results combination literature [1]. Letter alternatives themselves are obtained from the multiple interactive segmentation recognizer, which allows the reuse of partial results. The letter verification can be compared to the word ending postulation [6]. However, the word ending postulation is capable of introducing new results alternatives while the letter verification can only filter the already obtained results.

The use of the vertical bars recognizer together with the multiple interactive segmentation recognizer in a hybrid system provides a significant improvement of the recognition results. The resulting hybrid recognizer inherits some of the disambiguation problems from the vertical bars recognizer. The difference in recognition rates for top word alternative and larger numbers of alternatives is significant. Work is in progress to provide additional feature verification tools that would be used along with the letter verification. The objective is to build a system composed of small feature tools that push the correct word alternative to the top of the list.

References

[1] J. Franke. Statistical combination of multiple classifiers adapted on image parts. First European Conference on Postal Technology JET POSTE 93, France, pp. 566-572, 1993.

[2] Sh.A. Guberman, V.V. Rozentsveig. Algorithms for reading of handwritten texts. Avtomatika i Telemekhanika, No. 5, pp. 122-129, 1976 (in Russian)

[3] W. Guerfali, R. Plamondon. Normalizing and restoring on-line handwriting. Pattern Recognition, Vol. 26, No. 3, pp. 419-431, 1993

[4] B. Plessis, A. Sicsu, L. Heutte, E. Menu, E. Lecolinet, O. Debon, J.-V. Moreau. A multi-classifier combination strategy for the recognition of handwritten cursive words. Second Int. Conf. on Document Analysis and Recognition ICDAR'93, Japan, pp. 642-645, 1993

[5] R.K. Powalka, N. Sherkat, L.J. Evett, R.J. Whitrow. Multiple word segmentation with interactive look-up for cursive script recognition. Second Int. Conf. on Document Analysis and Recognition ICDAR'93, Japan, pp. 196-199, 1993

[6] R.K. Powalka, N. Sherkat, L.J. Evett, R.J. Whitrow. Dynamic cursive script recognition. A hybrid approach. in Advances in Handwriting and Drawing: A multidisciplinary approach, C. Faure, P. Keuss, G. Lorette, A. Vinter (eds.), pp. 137-154, 1994

[7] R.K. Powalka, N. Sherkat, R.J. Whitrow. The use of word shape information for cursive script recognition. Fourth Int. Workshop on Frontiers of Handwriting Recognition, Taiwan, pp. 67-76, 1994

[8] L. Xu, A. Krzyzak, C.Y. Suen. Methods for combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 418-435, 1992

[9] L.A. Zadeh. Fuzzy sets. Inform. Contr., Vol. 8, pp. 574-591, 1965